CHAPTER 11

NETWORKING WITH TCP/IP AND THE INTERNET

After reading this chapter and completing the exercises, you will be able to:

- Discuss additional details of TCP/IP addressing and subprotocols
- Comprehend the purpose and procedure for subnetting
- Understand the history and uses of BOOTP, DHCP, WINS, DNS, and host files
- Employ multiple TCP/IP utilities for network troubleshooting
- Understand TCP/IP applications, such as Internet browsers, e-mail, and voice over IP

ON THE JOB

My company, which designs custom-made bicycle frames, was one of the first companies on the block to use the Internet for e-mail. It allowed us to economically exchange information with our parts manufacturers and our distributors around the nation. As our company grew, and we opened a second office, we decided it would be great if we could use an intranet on our internal LAN to share information about orders, schedules, and budgets and to hold forums between staff. This turned out to be a great way to bring staff together.

Still, we felt we could do more with TCP/IP technology. Just last year, with the help of a local consulting company, we made our foray into e-commerce and opened up shop on the Internet. Our Internet sales were slow to begin with, but as word of our site traveled, we began receiving orders from around the world. Now our monthly revenue from Internet sales surpasses sales from all other means (including sales from our storefront).

Because of our success with e-commerce, I plan to steer most of my expenditures in the next fiscal year toward developing and supporting our Web site.

Terry Voss
ZAST, Inc.

The Internet is fast becoming not only a means of communication, but also a means of global commerce, development, and distribution. Industries such as banking, manufacturing, and healthcare depend on the Internet for daily transactions, recordkeeping, and sales. Individuals, too, are increasingly relying on the Internet for purchasing and data-gathering operations.

In previous chapters, you learned that the Internet depends on the TCP/IP suite of protocols, as do a number of network operating systems. Because of the increasing popularity of the Internet, having TCP/IP expertise can pave the way to a lucrative, challenging, and rewarding career. Even if your organization doesn't connect to the Internet, you will probably need to master TCP/IP to manage your network competently. In Chapter 3, you learned about the basic uses of TCP/IP, as well as TCP/IP subprotocols, routing capabilities, and addressing schemes. You also learned that TCP/IP is a complex and highly customizable protocol. This chapter builds on these basic concepts, examining how TCP/IP networks are managed, maintained, secured, and analyzed. You will start by learning more about TCP/IP addressing.

Addressing and Name Resolution

As you learned in Chapter 2, nodes on a network have both logical and physical addresses. (To refresh your memory about addressing and the OSI Model, it may be helpful to review Chapter 2 now.) The physical, or MAC, address is a unique number assigned to a device's NIC by the manufacturer at the factory. It belongs to the Data Link layer of the OSI Model. In contrast, the logical address belongs to the Network layer of the OSI Model and depends on the networking protocol used for data transmission in the Network layer (IP versus IPX, for example). A network administrator must manage logical addresses to ensure that every node on a network can communicate with other nodes, a process known as IP addressing. This section briefly reviews (logical) IP addressing before turning to ways to manage IP addressing that can make data transmission more reliable and your job easier.

IP Addressing

Just as you have a unique street address so as to ensure reliable delivery of your bills, letters, and magazines, every device on a TCP/IP-based network has a unique IP address to ensure accurate delivery of data. Without the existence of IP addresses, data could not be routed between networks and devices. Like street addresses, IP addresses must adhere to certain conventions. The following IP addressing characteristics should look familiar:

- An IP address is 32 bits in size.
- Every IP address is grouped into four 8-bit octets.
- Octets are separated by decimal points.
- Valid octet numbers range from 0 to 255 and represent a binary address. For example, an octet with the value of 68 equals 01 00 01 00 in an 8-bit binary pattern.
- Each address consists of two parts: network and host. The network portion is common to all nodes on one network, whereas the host portion is unique to each device. For example, two devices on the same Class C network might

have the following IP addresses: 208.133.78.11 and 208.133.78.17. In this example, the network portion of the address for both devices is "208.133.78"; the host portion is ".11" for the first device and ".17" for the second device.

- The network portion of an address indicates whether the device belongs to a Class A, B, C, D, or E network.
- Some octet numbers are reserved for special functions. For example, an address whose numeric value is the highest value in that network (host bits all equal to one) is a broadcast address—that is, an address used to communicate simultaneously with all nodes on a network. Other reserved addresses include those with a first octet of 127; these addresses are used exclusively for loopback testing.

Given these conventions, an example of a valid IP address for a networked workstation, printer, or other device might be 123.45.67.89.

IP addresses can be assigned manually on each device or automatically for a group of devices by a service called Dynamic Host Configuration Protocol (DHCP). (You will learn more about DHCP later in this chapter.) Typically, a network that supports DHCP uses this protocol for all of its devices, except those such as Web servers, which must have the same IP address at all times so that clients can reliably connect to them. An address that is assigned manually is called a **static address**, because it does not change unless a network technician reconfigures the device. An address that is assigned automatically by a service such as DHCP is called a **dynamic address**, because it can change over time.

Regardless of whether IP addresses are assigned manually or automatically, the network administrator must ensure that IP addresses are assigned consistently according to a plan and within the boundaries of an organization's valid IP address range. Recall from Chapter 3 that ICANN is the central authority on Internet addresses and names in the United States. In order to connect to the Internet, you must first obtain a group of IP addresses for your network from ICANN (or from an Internet service provider who has obtained them from ICANN).

Suppose that you are the network administrator for a company of 100 employees located in three separate offices (Downtown, East, and West) and that ICANN has assigned the range of IP addresses from 166.22.120.1 to 166.22.120.254 to your company. In total, you have 255 addresses to manage. If you have fewer than 10 servers, you might choose to assign the numbers 166.22.120.2 through 166.22.120.9 to your servers (as you'll learn later in this chapter, IP addresses ending in .1 are typically reserved for gateways). If you later analyze your network for performance or errors, you will know that a single digit in the last octet identifies a server node. You might then use DHCP to automatically assign addresses 166.22.120.10 through 166.22.120.99 to devices at the Downtown office, 166.22.120.100 through 166.22.120.199 to devices at the East office, and 166.22.120.200 through 166.22.120.254 to devices at the West office. Now, whenever you discover errors with a workstation whose last octet begins with a 2, you know that

you must examine the West office's network. This example illustrates merely one way that IP addresses can be managed to ease configuration and troubleshooting. Your assignment method will depend largely on the size of your network, its geographic scope, and the preferences of your technical staff.

The IP addresses given in the previous example (for example, 166.22.120.100) were expressed in dotted decimal notation. **Dotted decimal notation**, the most common way of expressing IP addresses, refers to the "shorthand" convention used to represent IP addresses and make them more easily readable by people. In dotted decimal notation, a decimal number between 1 and 255 represents each binary octet (a total of 256 possibilities). A period, or dot, separates each decimal. An example of a dotted decimal IP address is 10.65.10.18. Each number in the address has a binary equivalent, which is readable by the devices on the network. The binary value for 10.65.10.18, for example, is 00001010 01000001 00001010 00010010. You can easily calculate the binary value for a dotted decimal IP number by using the binary conversion feature on the calculator that comes with any Windows-based operating system (as described in Chapter 3).

Although you will most often use the dotted decimal notation in configuring and troubleshooting networks, to understand TCP/IP design issues such as subnetting (discussed later in this chapter), you must understand the binary foundation of IP addressing. Part of that binary foundation includes the network class to which each IP address belongs.

Network Classes

In Chapter 3, you learned that most IP addresses belong to one of three network classes—A, B, or C—and that the network class octets identify the network segment to which a device is attached. For example, your organization might have 20 workstations connected to one hub. All of these workstations would belong to the same network class.

A portion of each IP address contains clues about the network class. For example, an IP address whose first octet is in the range of 1–126 belongs to a Class A network. All IP addresses for devices on a Class A segment share the same first octet, or bits 0 through 7, as shown in Figure 11-1. The second through fourth octets (bits 8 through 31) in a Class A address identify the host.

An IP whose first octet is in the range of 128–191 belongs to a Class B network. All IP addresses for devices on a Class B segment share the first two octets, or bits 0 through 15. The third and fourth octets (bits 16 through 31) on a Class B network identify the host, as shown in Figure 11-1.

An IP address whose first octet is in the range of 192–223 belongs to a Class C network. All IP addresses for devices on a Class C segment share the first three octets, or bits 0 through 23. The fourth octet (bits 24 through 31) on a Class C network identifies the host, as shown in Figure 11-1. If your organization obtains its IP addresses from an Internet service provider (ISP), rather than directly from the ICANN, you probably use Class C addresses.

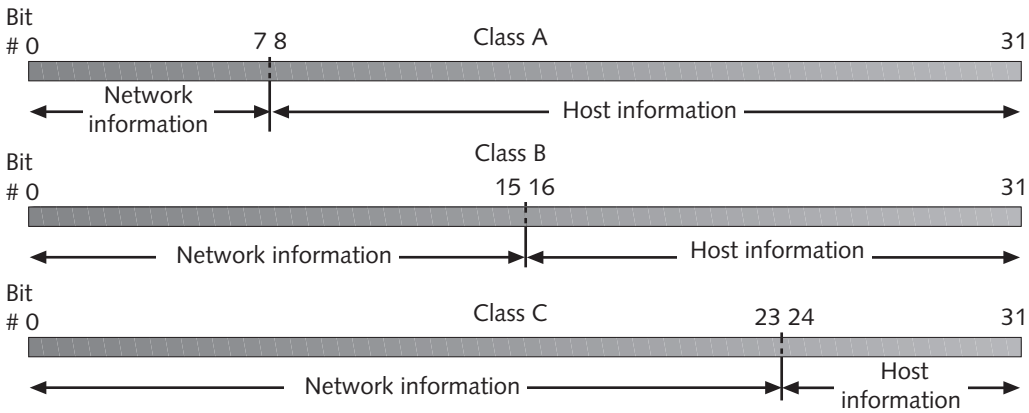

Figure 11-1 IP addresses and their classes

Chapter 3 also explained that each network class has a different number of networks and IP addresses available. For example, because Class A networks must begin with a number between 1 and 126, only 126 Class A networks exist in the world. More host names can be assigned to Class A networks, however, giving a total of more than 16 million possible addresses per network. Table 11-1 reviews what you have learned about the number of networks and addresses that belong to each class of network.

Table 11-1 The three commonly used classes of TCP/IP networks

Network Class	Beginning Octet	Number of Networks	Host Addresses per Network
A	1–126	126	16,777,214
B	128–191	> 16,000	65,534
C	192–223	> 2,000,000	254

In addition to Class A, B, and C networks, Class D and Class E networks also exist, but consumers and companies do not use them. Class D addresses, which begin with an octet whose value is between 224 and 239, are reserved for a special type of transmission called multicasting. **Multicasting** allows one device to send data to a specific group of devices (not the entire network segment). Whereas most data transmission is on a point-to-point basis, multicasting is a point-to-multipoint method. It can be used for teleconferencing or videoconferencing over the Internet, for example. The Internet Engineering Task Force (IETF) reserves Class E addresses, which begin with an octet whose value is between 240 and 254, for experimental use. You should never use Class D or Class E addresses when configuring your network.

You may think that the use of network classes automatically provides easy organization and a sufficient quantity of IP addresses on the Internet. Although this goal is what the Internet's founders intended, it hasn't necessarily come to pass. In the early days of the Internet, Class A addresses were distributed liberally, with some organizations receiving

more reserved addresses than they had devices. Today, many addresses on the Internet go unused, but cannot be reassigned. In addition, although potentially more than 4.3 billion Internet addresses are available, the demand for such addresses grows exponentially every year. The early designers of Internet addressing did not anticipate this kind of growth. To respond to this demand, a new addressing scheme is being developed that can supply the world with enough addresses to last well into the twenty-first century. **IP version 6 (IPv6)**, also known as the next-generation IP, will incorporate this new addressing scheme.

The Web contains a wealth of information about the Internet, including statistics on the number of currently used IP addresses. One example of such a repository is found at Internet Software Consortium's site at *www.isc.org/ds/*. The Internet Software Consortium estimates the number of hosts on the Internet by querying machines worldwide and compiling records of Internet address assignments. It also offers links to graphs that depict the Web's growth over the last decade. Another good site for information on Internet usage, developments, and standards information is hosted by the Internet Sciences Institute at the University of Southern California and can be found at *info.internet.isi.edu/1/in-notes/*.

Subnetting

Subnetting is the process of subdividing a single class of network into multiple, smaller networks. Because it results in a more efficient use of IP addresses, subnetting was implemented throughout the Internet in the mid-1980s. Before subnetting, each segment on a network required its own Class A, B, or C network number. With this scheme, if you used a 10Base2 network with a 30-node limitation, once you exceeded 30 devices on your network, you would have needed to request another class of addresses from ICANN (or at the time, InterNIC). As you can imagine, this approach was not an efficient use of IP addresses or network managers' time. Not only did a network manager have to apply (and pay) for a new class of addresses, but he or she also needed to change the network's routing tables to accommodate each new network class.

With subnetting, however, a network manager can use one class of addresses for several network segments. This approach becomes possible because one of the address's octets is used to indicate how the network is subdivided, or subnetted. Rather than consisting simply of network and host information, a subnetted address includes network, subnet, and host information, as shown in Figure 11-2. For example, under normal circumstances, if ICANN granted your organization all of the IP addresses that shared the Class B network ID of 166.144, the last two octets would be available for host information. To better organize your addresses and allow for growth, however, you would be wise to devote the third octet to subnet information. By using the third octet to subdivide the network, you create the functional equivalent of 254 Class C networks (166.144.0.0 through 166.144.254.0) from your single Class B network. As far as ICANN is concerned, you continue to use a Class B network; within your organization, on the other hand, your LAN is fooled into recognizing several Class C networks.

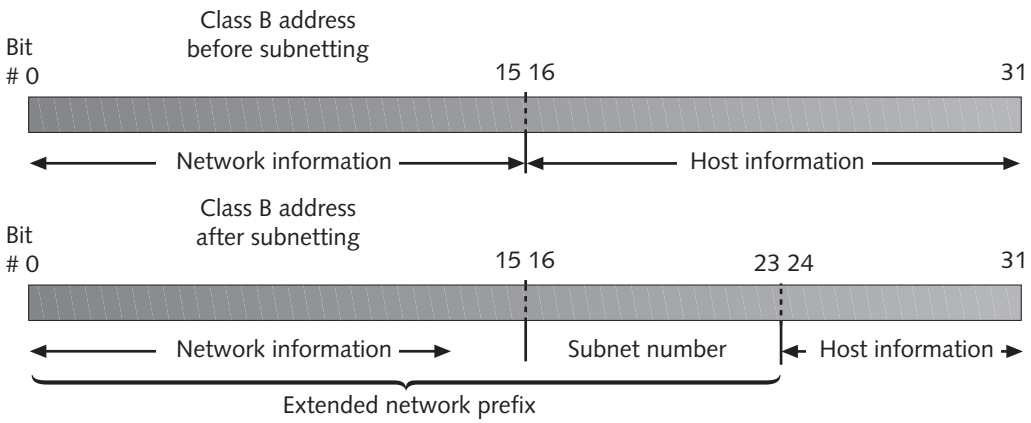

Figure 11-2 IP addresses before and after subnets

The combination of an address's network and subnet information constitutes its extended network prefix. By interpreting an address's **extended network prefix**, a device can determine the subnet to which an address belongs. But how does the device know whether an address is part of a subnet in the first place? After all, the third octet in an IP address could be either a Class B address's subnet number or a part of a Class B address's network information. For example, how can a device tell whether the address 166.144.40.33 belongs to a Class B subnetted network or a Class B network that has not been subnetted?

To make this determination, the device interprets a subnet mask. A **subnet mask** is a special 32-bit number that, when combined with a device's IP address, informs the rest of the network about the network class to which the device is attached. Subnet masks are specified in the same way that IP addresses are specified—either manually, within a device's TCP/IP configuration, or automatically, through a service such as DHCP.

Subnet masks, like IP addresses, are composed of four octets and can be expressed in either binary or dotted decimal notation. An octet of all 1s (using binary notation) in a subnet mask represents part of the extended network prefix in a subnetted IP address that uses that subnet mask. Otherwise, the subnet mask bits are all 0s, and the corresponding octets in the subnetted IP address that uses that subnet mask are assumed to represent host information. Thus, for the subnetted IP address 166.144.40.33, the subnet mask would be 11111111 11111111 11111111 00000000 in binary notation or 255.255.255.0 in dotted decimal notation. In this example, the first three octets make up the extended network prefix. Figure 11-3 shows the correlation between an IP address and its subnet mask.

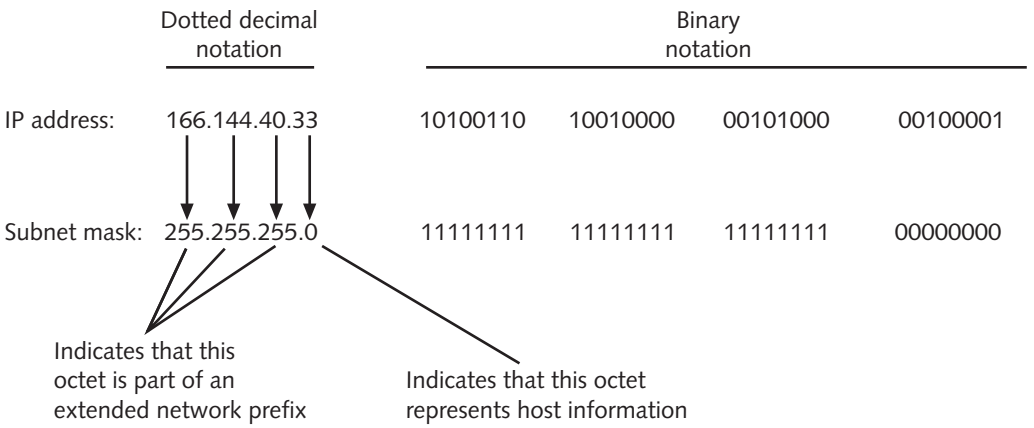

Figure 11-3 A subnetted IP address and its subnet mask

If you do not use subnetting, the extended network prefix will simply equal the network portion of the IP address. You might not want to use subnetting in only three situations: (1) if you have a very small network (as in your home office), (2) if you never want to connect to the Internet and so do not have to follow IP addressing standards, or (3) if your organization has more IP addresses than it can ever conceivably use. If you don't specify a subnet mask, the default subnet mask is 255.0.0.0 for a Class A network, 255.255.0.0 for a Class B network, and 255.255.255.0 for a Class C network. To qualify for Net+ certification, you should be familiar with the default subnet masks associated with each network class.

If you use subnetting on your LAN, only your LAN's devices need to interpret your devices' subnetting information. Routers external to your LAN, such as those on the Internet, pay attention to only the network portion of your devices' IP addresses when transmitting data to them. Recall that the network portions of an organization's addresses are not affected by subnetting and that subnetting modifies only octets that would otherwise be used for host information. As a result, devices external to a subnetted LAN (such as routers on the Internet) can direct data to those LAN devices without interpreting the LAN's subnetting information.

Figure 11-4 illustrates a situation in which a LAN has been granted the Class B range of addresses that begin with 166.144. The network administrator has subnetted this Class B network into at least six smaller networks that begin with the following Class C network prefixes: 166.144.40, 166.144.42, 166.144.56, 166.144.59, 166.144.60, and 166.144.63. When a router on the internal LAN needs to direct data from a machine with the IP address of 166.144.40.12 to a machine with the IP address of 166.144.60.12, its interpretation of the workstations' subnet masks (255.255.255.0) tells the router that they are on different subnets. When a server on the Internet attempts to deliver a Web page to the machine with IP address 166.144.40.12, however, the Internet router does not use the subnet mask information but rather assumes that the machine is on a Class B network. That's all the information it needs to know to reach the organization's router. Once the data enter the organization's LAN, the LAN's router then interprets the subnet mask information as if it were transmitting data

internally to deliver data to the machine with IP address 166.144.40.12. Because subnetting does not affect how a device is addressed by external networks, a network administrator does not need to inform Internet authorities about new networks created via subnets.

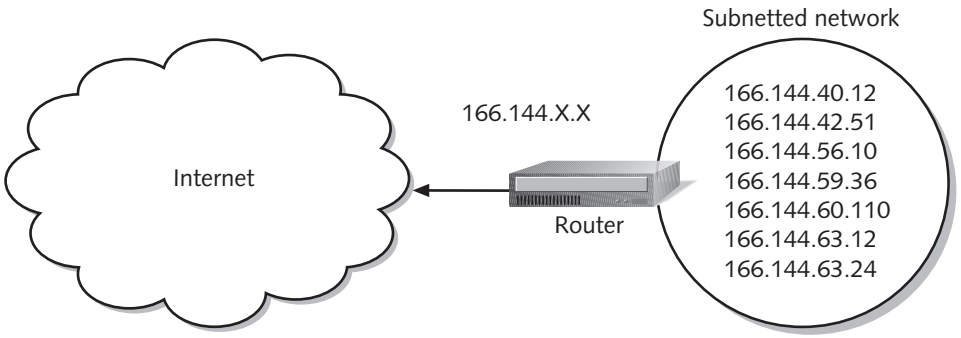

Figure 11-4 A subnetted network connected to the Internet

As you know, routers connect different network segments via their physical interfaces. In the case of subnetting, a router must interpret IP addresses from different subnets and direct data from one subnet to another. Each subnet corresponds to a different interface on the router, and each interface is associated with a default gateway address (described later in this chapter) that ends with "1". Figure 11-5 depicts a network with several subnets connected through a router.

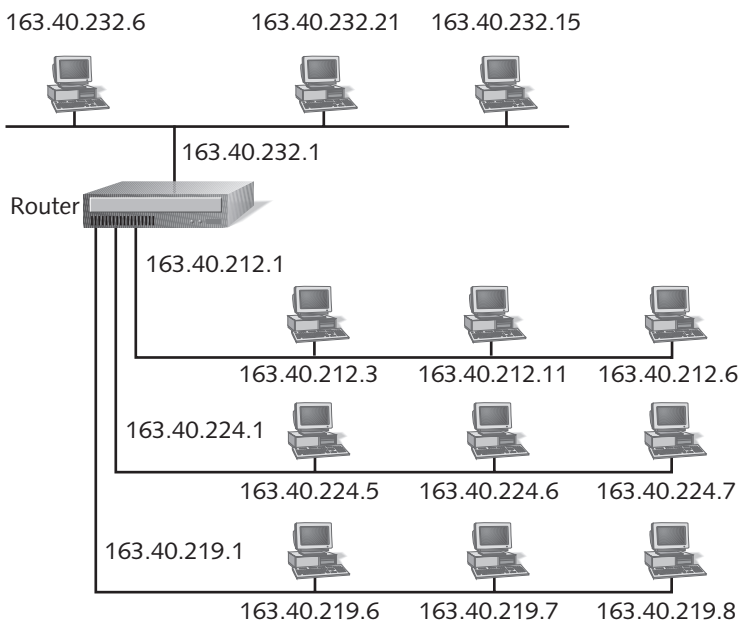

Figure 11-5 A network with several subnets

To understand how subnets and IP addresses work on a network, it's helpful to follow the (simplified) path of data between nodes on different subnets. Imagine you are sitting at the Marketing PC in Figure 11-6. Your colleague has asked you to retrieve a research report from the Internet and then print it for her. The closest printer is located in the IT department, which resides on a different subnet.

First, you connect to the Internet and request the report from the marketing research Web site. Your network's router processes your request, determining that the IP address of the Web site is on the Internet, and passes your request to the Internet routers. The Internet routers forward your request to the marketing research Web site by reading the network information in the destination IP address. The marketing research Web site's server interprets your request for data and sends the report back to the Internet routers, with your IP address in the destination portion of the header. The routers on the Internet read only the network portion of your IP address and forward the data to your router. Your router analyzes the extended network prefix in the destination address to discover your subnet, then uses the host information to send the data to your machine.

When you choose to print the report, your network router interprets your request. It reads the printer's IP address, recognizes that it is a subnetted address, and looks at its extended network prefix to find out the subnet to which it belongs. Then your router forwards the request to the router interface that services the printer's subnet (it may be on the same router or another router). The request proceeds to the printer.

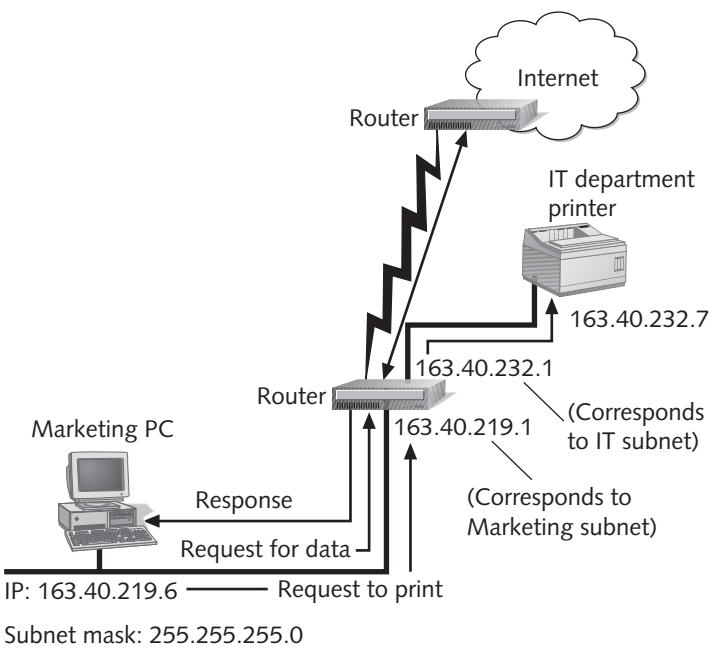

Figure 11-6 Data traveling over subnets

Gateways

In Chapter 6, you learned that **gateways** are a combination of software and hardware that enable two different network segments to exchange data. In the context of IP addressing, a gateway facilitates communication between different subnets. Because one device on the network cannot send data directly to a device on another subnet, a gateway must intercede and hand off the information. Every device on a TCP/IP-based network has a **default gateway**—that is, the gateway that first interprets its outbound requests to other subnets, and then interprets its inbound requests from other subnets.

A gateway is analogous to your local post office. Your post office gathers your outbound mail and decides where to forward it. It also handles your inbound mail just before it heads for your mailbox. Just as a large city has several local post offices, a large organization will have several gateways to route traffic for different groups of devices. Each node on the network can have only one default gateway; that gateway is assigned either manually or automatically (in the latter case, through a service such as DHCP). Of course, if your network includes only one segment and you do not connect to the Internet, your devices would not need a default gateway because traffic would not need to cross the network's boundary.

In many cases a default gateway is not a separate device, but rather a network interface on a router. In this way, one router can supply multiple gateways. Each default gateway is assigned its own IP address. In Figure 11-7, workstation 10.3.105.23 (workstation A) uses the 10.3.105.1 gateway to process its requests, and workstation 10.3.102.75 (workstation B) uses the 10.3.102.1 gateway for the same purpose.

An IP gateway is usually assigned an IP address that ends with an octet of .1.

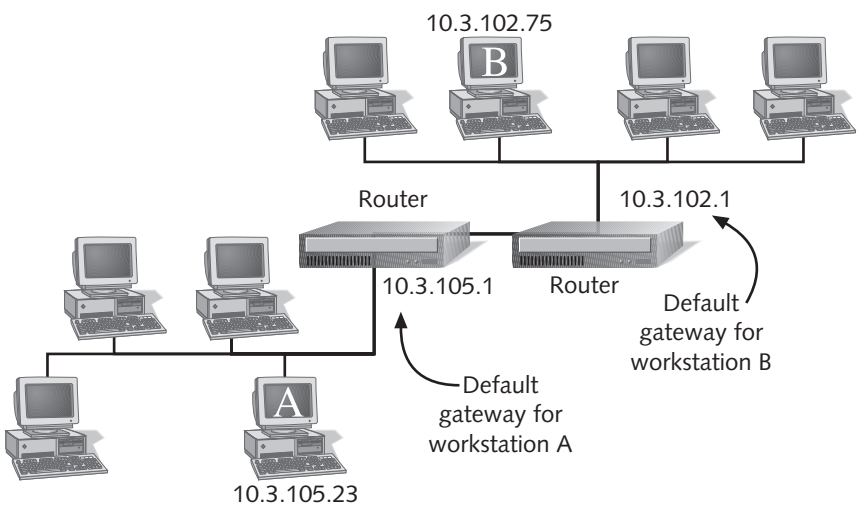

Figure 11-7 The use of default gateways

Default gateways may connect multiple internal networks, or they may connect an internal network with external networks such as WANs or the Internet. As you learned in Chapter 6, routers that connect multiple networks must maintain a routing table to determine where to forward information. When a router is used as a gateway, it must maintain routing tables as well.

The Internet contains a vast number of routers and gateways. If each gateway had to track addressing information for every other gateway on the Internet, it would be overtaxed. Instead, each handles only a relatively small amount of addressing information, which it uses to forward data to another gateway that knows more about the data's destination. Like routers on an internal network, Internet gateways maintain default routes to known addresses to expedite data transfer. The gateways that make up the Internet backbone, called **core gateways**, are managed by the Internet Network Operations Center (INOC).

Network Address Translation (NAT)

IP gateways can also be used to "hide" the IP numbers assigned within an organization and keep its devices' IP addresses secret from any public network (such as the Internet). Not only can hiding the IP addresses protect their identity, but it can also allow network managers more flexibility in assigning addresses. Clients behind the gateway may use any IP addressing scheme, whether or not it is legitimately recognized by the Internet authorities. But once those clients need to connect to the Internet, they must have a legitimate IP address in order to exchange data. When the client's transmission reaches the IP gateway, the gateway assigns the client's transmission a valid IP address. This process is known as **network address translation (NAT)**.

One reason for hiding IP addresses is to add a marginal amount of security to a private network when it is connected to a public network (such as the Internet). Because your transmission is assigned a new IP address when it reaches the public sphere, others outside your organization cannot trace the origin of your transmission.

Another reason for using NAT is to enable a network administrator to develop her own network addressing scheme that does not conform to a scheme dictated by ICANN. For example, suppose you are the network administrator for a private elementary school. You maintain the school's entire network, which, among other things, includes 200 client workstations. Suppose half of these clients are used by students in the classrooms or library and half are used expressly by staff. In order to make your network management easier, you might decide to assign each student workstation an IP address whose first octet begins with the number 10 and whose second octet is the number of the classroom where the computer is located. (For example, a student workstation in room 235 might have an IP address of 10.235.1.12.) You might then assign each staff workstation an IP address whose first octet is the number 50 and whose second octet is the number of the employee's office or classroom. (For example, the principal's workstation, which is located in his office in Room 110, might have an IP address of 50.110.1.10.) These IP addresses

would be used strictly for communication between devices on the school's network. When staff or students wanted to access the Internet, however, you would need to have at least some IP addresses that would be legitimate for use on the Internet. Now suppose that, because the school has limited funds and does not require that all clients be connected to the Internet at all times, you decide to purchase a block of only 20 IP numbers and set up an IP gateway to translate your internal addresses to addresses that can be used on the Internet. Each time a client attempts to reach the Internet, the IP gateway would replace its source address field in the data packets with one of the 20 legitimate IP addresses. Figure 11-8 depicts how the process of NAT works.

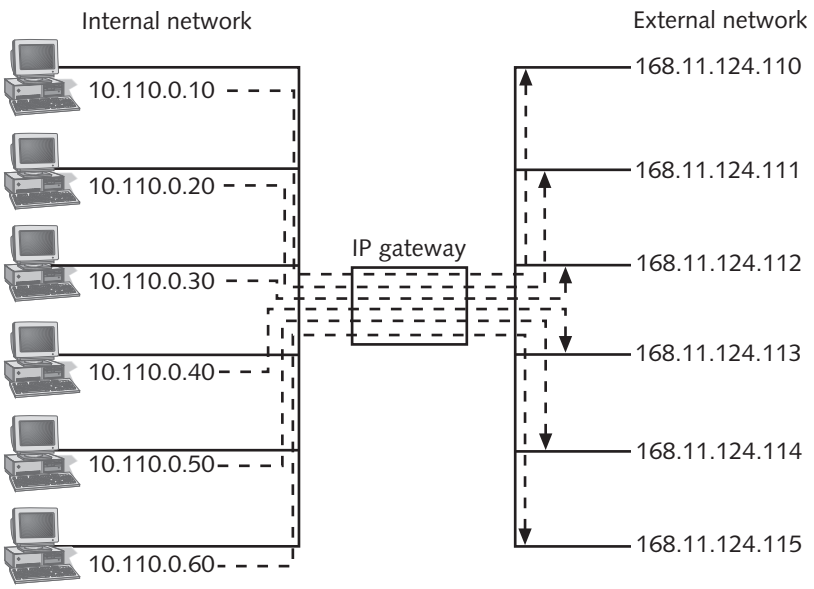

Figure 11-8 NAT through an IP gateway

Sockets and Ports

In Chapter 3, you learned that a **socket** is a logical address assigned to a specific process running on a host computer. It forms a virtual connection between the host and client. The socket's address combines the host computer's IP address with the **port number** associated with a process. For example, the Telnet service on a Web server with an IP address of 10.43.3.87 might have a socket address of 10.43.3.87:23, where 23 is the standard port number for the Telnet service. In other words, after installation, the Web server software assumes that any requests coming into port number 23 are Telnet requests, unless you configure the software differently. Note that a port number is expressed as a number following a colon after an IP address. The 23 is not considered an additional octet in the socket number, but simply a pointer to that port.

Port numbers can have any value. Some software programs that use TCP/IP (for example, Novell's GroupWise and Hewlett-Packard's Performance Data Alarm Manager) choose their own port numbers by default. The default port numbers for commonly used TCP/IP services generally have values lower than 255, as shown in Table 11-2. Port numbers in the range of 0 to 1023 are also called **well-known ports**, because they were long ago assigned by Internet authorities to popular services (for example, FTP and Telnet), and are therefore well known and frequently used.

Although you do not need to memorize every port number for the Net+ Certification exam, you may be asked about the port numbers associated with common services, such as Telnet, FTP, SNMP, and HTTP. Knowing them will also help you in configuring and troubleshooting TCP/IP networks.

Table 11-2 Commonly used TCP/IP port numbers

Port Number	Process Name	Protocol Used	Description
1	TCPMUX	TCP	TCP Port Multiplexer Service
5	RJE	TCP	Remote job entry
7	ECHO	TCP and UDP	Echo
11	USERS	TCP and UDP	Active users
13	DAYTIME	TCP and UDP	Daytime
17	QUOTE	TCP and UDP	Quote of the Day
20	FTP-DATA	TCP	File transfer - data
21	FTP	TCP	File transfer - control
23	TELNET	TCP	Telnet
25	SMTP	TCP	Simple Mail Transfer Protocol
35	Printer	TCP and UDP	Any private printer service
37	TIME	TCP and UDP	Time
41	GRAPHICS	TCP and UDP	Graphics
42	NAMESERV	UDP	Host name server
43	NICNAME	TCP	Who is
49	LOGIN	TCP	Login Host Protocol
53	DNS	TCP and UDP	Domain name server
67	BOOTPS	UDP	Bootstrap Protocol server
68	BOOTPC	UDP	Bootstrap Protocol client
69	TFTP	UDP	Trivial File Transfer Protocol
79	FINGER	TCP	Finger
80	HTTP	TCP and UDP	World Wide Web HTTP
101	HOSTNAME	TCP and UDP	NIC host name server
105	CSNET-NS	TCP and UDP	Mailbox name server
110	POP3	TCP	Post Office Protocol 3

Table 11-2 Commonly used TCP/IP port numbers (continued)

Port Number	Process Name	Protocol Used	Description
119	NNTP	TCP and UDP	Network News Transport Protocol
137	NETBIOS-NS	TCP and UDP	NETBIOS Name Service
138	NETBIOS-DG	TCP and UDP	NETBIOS Datagram Service
139	NETBIOS-SS	TCP and UDP	NETBIOS Session Service
161	SNMP	UDP	Simple Network Management Protocol
162	SNMPTRAP	UDP	SNMPTRAP
179	BGP	TCP	Border Gateway Protocol

The use of port numbers simplifies TCP/IP communications and ensures that data are transmitted to the correct application. When a client requests communications with a server and specifies port 23, for example, the server knows immediately that the client wants a Telnet session. No extra data exchange is necessary to define the session type, and the server can initiate the Telnet service without delay. The server will connect to the client's Telnet port—by default, port 23—and establish a virtual circuit. Figure 11-9 depicts this process.

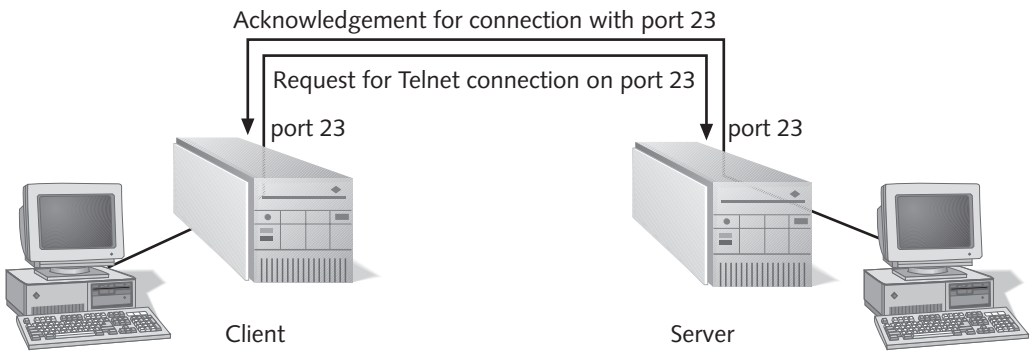

Figure 11-9 A virtual circuit for the Telnet service

As mentioned earlier, you can configure port numbers through software. Most servers maintain an editable, text-based file of port numbers and their associated services. If necessary, you could change the default port number for the Telnet service on your server from 23 to 2330. Changing a default port number is rarely a good idea, however, because it violates the standard. Nevertheless, some network administrators who are preoccupied with security may change their servers' port numbers in an attempt to confuse potential hackers.

Host Names and Domain Name System (DNS)

As you have seen, much of TCP/IP addressing involves numbers—often long, complicated numbers. Computers can manage numbers easily. However, most people can remember words better than numbers. Imagine if you had to identify your friends' and families' Social Security numbers whenever you wanted to write a note or talk to them. Communication would be frustrating at the very least, and perhaps even impossible—especially if you're the kind of person who has trouble remembering even your own Social Security number. Similarly, people prefer to associate names with networked devices rather than remember IP addresses. For this reason, the Internet authorities established a naming system for all nodes on the Internet.

As you learned in Chapter 3, every device on the Internet is technically known as a host. Every host can take a **host name**, a name that describes the device. For example, someone named Peggy Sue McDonald might name her workstation "PeggySue." If the computer is reserved for a specific purpose, you may want to name it accordingly. For example, a company that offers free software downloads through the FTP service might call its host machine "ftpserver." Often, when networking professionals refer to a machine's host name, they mean its local host name plus its domain name—in other words, its fully qualified host name. The following sections discuss host names and domain names.

Domain Names

Every host is a member of a domain, or a group of computers that belong to the same organization and have part of their IP addresses in common. A domain is identified by its domain name. Usually, a **domain name** is associated with a company or other type of organization, such as a university or military unit. For example, IBM's domain name is ibm.com, and the U.S. Library of Congress's domain name is loc.gov. If you worked at the Library of Congress and had named your workstation "PeggySue," your full host name (also known as your fully qualified host name) might be "PeggySue.loc.gov."

Although an individual user can name his or her workstation, organizations cannot arbitrarily choose their own domain names. Domain names must be registered with the Internet naming authority, ICANN. If you use an ISP, your ISP can work with ICANN to obtain a domain name for you, providing that someone else hasn't already reserved your desired name.

ICANN has established conventions for domain naming in which certain suffixes apply to every type of organization that uses the Internet. These suffixes are also known as **top-level domains (TLDs)**. Table 11-3 lists common TLDs. The first eight TLDs listed in this table were established in the mid 1980s. In the past few years, organizations have appealed to ICANN to add the remaining seven TLDs, some of which have been approved, and some of which are still pending approval. In addition, each country has its own domain suffix. For example, Canadian domains end with .ca and Japanese domains end in .jp.

Although domain names may use the international domain suffix, such as .ca for Canada and .jp for Japan, organizations do not necessarily have to use these suffixes. For example, although IBM's headquarters are located in the United States, the company's domain name is *www.ibm.com*. On the other hand, some U.S. organizations do use the .us suffix. For example, the domain name for the Garden City, New York, public school district is *www.gardencity.k12.ny.us*.

Table 11-3 Domain naming conventions

Domain Suffix	Type of Organization
ARPA	Reverse lookup domain (special Internet function)
COM	Commercial
EDU	Educational
GOV	Government
ORG	Non-commercial organization (such as a nonprofit agency)
NET	Network (such as an ISP)
INT	International Treaty Organization
MIL	U.S. military organization
BIZ	Businesses
INFO	Unrestricted use
AERO	Air-transport industry
COOP	Cooperatives
MUSEUM	Museums
NAME	Individuals
PRO	Professionals (such as doctors, lawyers, and engineers)

Once an organization reserves a domain name, the rest of the world's computers know to associate that domain name with that particular organization, and no other organization can legally use it (as long as the reserving organization pays the required annual registration fee to ICANN). For example, you might apply for the domain name called "YourName.com"; not only would the rest of the Internet associate that name with your machine, but also no other parties in the world could use "YourName.com" for their machines.

Host names come with some naming restrictions. You can use any alphanumeric combination with a maximum of 63 characters, and you can include hyphens, underscores, or periods in the name, but no other special characters. The interesting part of host and domain naming relates to how all Internet-connected machines in the world know which names belong to which machines. Before tackling the entire world, however, you can start by thinking about how one company might deal with its local host names.

Host Files

The first incarnation of the Internet (called ARPAnet) was used by fewer than 1000 hosts. The entire Internet relied on one text file called HOSTS.TXT to associate names with IP addresses. This file was generically known as a **host file**. The explosive growth of the Internet soon made this simple arrangement impossible to maintain—the host file would require constant changes, searching through one file from all over the nation would strain the Internet's bandwidth capacity, and the entire Internet would fail if the file were accidentally deleted.

Within a company or university, you may still encounter this older system of straightforward ASCII text files that associate internal host names with their IP addresses. Figure 11-10 provides an example of such a file. Notice that each host is matched by one line identifying the host's name and IP address. In addition, a third field, called an **alias**, provides a nickname for the host. An alias allows a user within an organization to address a host by a shorter name than the full host name. Typically, the first line of a host file begins with a pound sign and contains comments about the file's columns. A pound sign may precede comments anywhere in the host file.

```
# IP address      host name            aliases
132.55.78.109     bingo.games.com      bingo
132.55.78.110     parcheesi.games.com  parcheesi
132.55.78.111     checkers.games.com   checkers
132.55.78.112     darts.games.com      darts
```

Figure 11-10 An example of a host file

On a UNIX-based computer, a host file is called **hosts** and is located in the /etc directory. On a Windows 9x computer, it is called **lmhosts** and must be located in the c:\windows directory in order to be recognized by the operating system. On a Windows NT or Windows 2000 computer, the file may be called hosts or lmhosts, and must be located in the %systemroot%\system32\drivers\etc folder (where %systemroot% is the directory in which the operating system is installed). Each Windows operating system includes a sample lmhosts file called lmhosts.sam, which is a plain text file you can view in the Notepad program. If you are using hosts or lmhosts files, you should not only master the syntax of this file, but you should also research the implications of using a static host file on your network.

Domain Name System (DNS)

A simple host file can satisfy the needs of one organization, and it can even allow one organization's network to contact hosts on another network. A single host file is no longer sufficient for the Internet, however. Instead, a more automated solution has become mandatory. In the mid-1980s, the Network Information Center (NIC) at Stanford Research Institute devised a hierarchical way of tracking domain names and their addresses, called the **Domain Name System (DNS)**. The DNS database does not

rely on one file or even one server, but rather is distributed over several key computers across the Internet to prevent catastrophic failure if one or a few computers go down. DNS is a TCP/IP service that belongs to the Application layer of the TCP/IP Model.

In a simple world, the responsibility for resolving addresses to names might be broken down into national, regional, local, and organizational levels, as depicted in Figure 11-11. In this example, if you worked in the Atlanta office and wanted to exchange data with a machine at another company located in Fairbanks, your organizational DNS server would take your request and pass it off to the local server, which would then pass it off to the regional server, which would in turn pass it off to the national DNS server. The reverse process would occur as the national DNS server passed the request down to the regional, local, and finally organizational servers that know about Fairbanks. This address resolution process assumes that your local DNS server and the Fairbanks local DNS server do not know where to find each other.

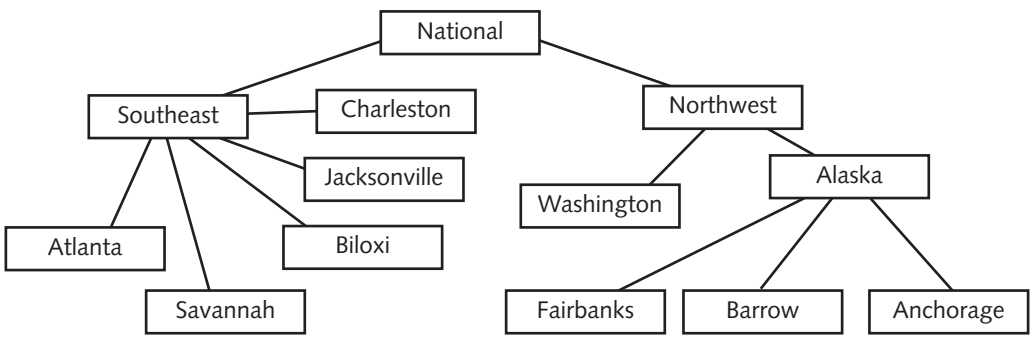

Figure 11-11 Simplified DNS server hierarchy by geography

The actual DNS is somewhat more sophisticated than the previous example implies. To route traffic more efficiently, it is divided into three components: resolvers, name servers, and name space. **Resolvers** are any hosts on the Internet that need to look up domain name information. In the example given in the previous paragraph, your machine in Atlanta is a resolver. The resolver client is built into TCP/IP applications such as Telnet, HTTP, and FTP. If you type the command telnet support.novell.com, your Telnet client software will kick off the resolver service to find the IP address for support.novell.com. If you have Telnetted to the site before, the information may exist in temporary memory and may be retrieved very quickly. Otherwise, the resolver service queries your machine's name server to find the IP address for support.novell.com.

Name servers are servers that contain databases of names and their associated IP addresses. A name server supplies a resolver with the information it requires. If the name server cannot resolve the IP address, the query passes to a higher-level name server. In Figure 11-11, the local, regional, and national servers are all name servers. Each name server manages a group of devices, collectively known as a **zone**; these devices, in turn, distribute naming information. In the example in Figure 11-11, the Southeast name server's zone would include the Atlanta, Savannah, Jacksonville, Biloxi, and Charleston

name servers. If the Atlanta server doesn't know the address of a machine in the Savannah area, it can rely on the Southeast name server to supply that information. In a small company, the primary DNS server's zone would include all the computers at the company.

Configuring DNS

Any host that must communicate with other hosts on the Internet needs to know how to find its name server. Although some organizations use only one name server, large organizations often maintain two name servers—a primary and a secondary name server—to help ensure Internet connectivity. If the primary name server experiences a failure, all devices on the network will attempt to use the secondary name server. Each device on the network relies on the name server and therefore must know how to find it. When configuring the TCP/IP properties of a workstation, you need to specify a name server IP address so that the workstation will know which machine to query when it needs to look up a name.

To view or change the name server information on a Windows 2000 workstation:

1. Click **Start**, point to **Settings**, point to **Control Panel**, and then click **Network and Dial-up Connections**. The Network and Dial-up Connections window opens.
2. Right-click the **Local Area Connection** icon and click **Properties** in the shortcut menu. The Local Area Connection Properties dialog box appears.
3. Click **Internet Protocol (TCP/IP)** in the list of network components, and then click **Properties**. The Internet Protocol (TCP/IP) Properties dialog box opens, as shown in Figure 11-12.

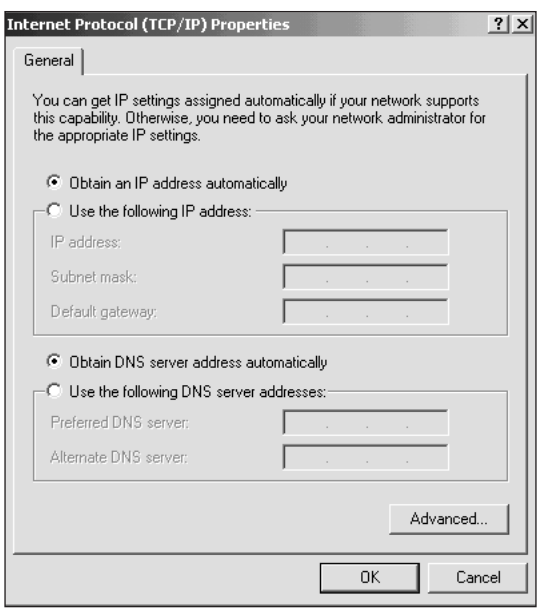

Figure 11-12 Internet Protocol (TCP/IP) Properties dialog box in Windows 2000

4. If necessary, click the **Use the following DNS server addresses** option button to select it.
5. Enter the IP address of your primary DNS server in the Preferred DNS server text box.
6. Enter the IP address of your secondary DNS server, if you have one, in the Alternate DNS server text box.
7. Click **OK** to close the Internet Protocol (TCP/IP) Properties dialog box and save your changes.
8. Click **OK** to close the Local Area Connection Properties dialog box.

To view or change the name server information on a Windows 9x machine:

1. Right-click the **Network Neighborhood** icon, and then click **Properties** in the shortcut menu. The Network Properties dialog box opens.
2. In the list of installed network components, double-click the TCP/IP protocol that is bound to your network adapter. The TCP/IP Properties dialog box opens.
3. Click the **DNS Configuration** tab. The DNS Configuration tab appears, as shown in Figure 11-13.

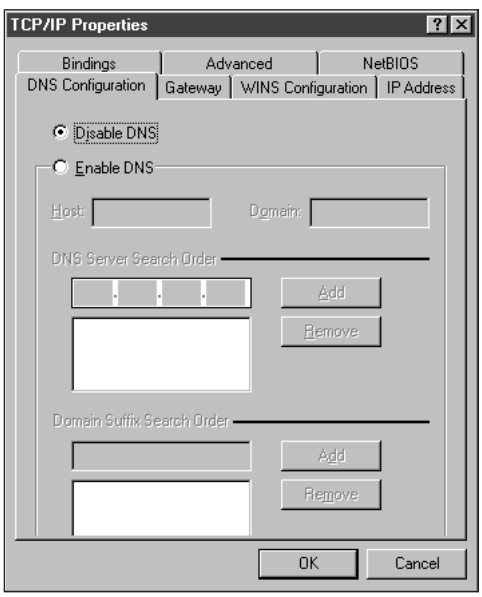

Figure 11-13 DNS Configuration properties tab

4. Click the **Enable DNS** option button to select it (unless you are using DHCP or, in some cases, dial-up networking).
5. Type your computer's host name in the Host text box.

6. Type your organization's domain name in the Domain text box.
7. Type your organization's DNS server IP address in the space provided under the heading "DNS Server Search Order."
8. Click **Add** to save the DNS server's IP address.
9. Add as many as two more DNS server IP addresses in the same manner.
10. Click **OK** to save your changes.
11. Click **OK** to close the Network Properties dialog box.
12. Click **Yes** to confirm that you want to restart your computer.

DNS Name Space

As you learned previously, many name servers across the globe cooperate to keep track of IP addresses and their associated domain names. **Name space** refers to the actual database of Internet IP addresses and their associated names. Every name server holds a piece of the DNS name space. At the highest level in the hierarchy sit the root servers. A **root server** is a name server that is maintained by ICANN and that acts as the ultimate authority on how to contact the top-level domains, such as those ending with .com, .edu, .net, .us, and so on. ICANN maintains 13 root servers around the world. In Figure 11-11, the national-level name server would actually be a root server for the United States.

Name space is not a database that you can open and view like a store's inventory database. Rather, this abstract concept describes how the name servers of the world share DNS information. Pieces of it are tangible, however, and are stored on a name server in a **resource record**, which is a single record that describes one piece of information in the DNS database. For example, an **address resource record** is a type of resource record that maps the IP address of an Internet-connected device to its domain name.

Resource records come in many different types, depending on their function. Each resource record contains a name field to identify the domain name of the machine to which the record refers, a type field to identify the type of resource record involved, a class field to identify the class to which the record belongs (usually "IN" or "Internet"), a time to live field to identify how long the record should be saved in temporary memory, a data length field to identify how much data the record contains, and the actual record data. Approximately 20 types of resource records are currently used.

Each resource record type adheres to specific data field requirements, thus ensuring that any name server across the world can interpret it. For example, the data field of a simple network address record would include only the network address, whereas the data field of a mailbox information record would include the name of the mailbox responsible for error messages and the name of the mailbox responsible for mailing lists. In the following fictitious address resource record, knight.chess.games.com is the host domain

name, IN stands for the Internet record class, A identifies the record type as "address," and 203.99.120.76 is the host's IP address:

```
knight.chess.games.com   IN   A    203.99.120.76
```

This book does not provide in-depth coverage of DNS domains, hierarchy, zones, and databases. If you are interested in Internet server administration, you should investigate host files and DNS in more detail. For Net+ certification, you should know the purpose of DNS and host files, understand the hierarchical nature of DNS, and be able to specify name servers on a client workstation.

BOOTP

To communicate with other devices through TCP/IP, every workstation, printer, or other node on a network requires a unique IP address. On the earliest TCP/IP networks, each device was manually assigned its own number through a local configuration file; that number never changed until someone edited the configuration file. As networks grew larger, however, local configuration files became more difficult to implement. Imagine the arduous task faced by a network administrator who must visit each of 8000 workstations, printers, and hosts on a company's LAN to assign IP addresses and ensure that no single IP address is used twice. Now imagine how much extra work would be required to restructure the company's IP address management system (for example, to implement subnetting) or to move a department's machine to a different network segment.

To facilitate IP address management, a service called the Bootstrap Protocol was developed in the mid-1980s. The **Bootstrap Protocol (BOOTP)** uses a central list of IP addresses and their associated devices' MAC addresses to dynamically assign IP addresses to clients. When a client that relies on BOOTP first connects to the network, it sends a broadcast message to the network asking to be assigned an IP address. This broadcast message includes the MAC address of the client's NIC. The BOOTP server recognizes a BOOTP client's request, looks up the client's MAC address in its BOOTP table, and responds to the client with the following information: the client's IP address, the IP address of the server, the host name of the server, and the IP address of a default router. Figure 11-14 outlines this process.

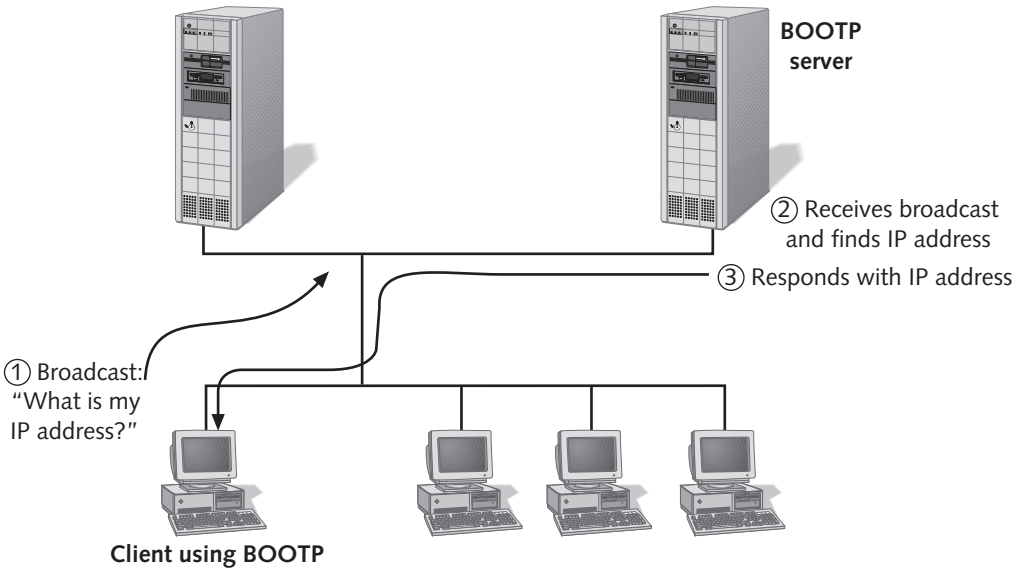

Figure 11-14 The BOOTP process

Thanks to BOOTP, a client does not have to remember its own IP address, and therefore network administrators do not have to go to each workstation on a network in order to manually assign its IP address. This situation is ideal for **diskless workstations** (that is, workstations that do not contain hard disks, but rely on a small amount of read-only memory to connect to a network and to pick up their system files).

For other kinds of clients, BOOTP has been surpassed by the more sophisticated IP address management tool, Dynamic Host Configuration Protocol (DHCP). As you will learn in the next section, DHCP requires little intervention, whereas BOOTP requires network administrators to enter every IP and MAC address manually into the BOOTP table. As you can imagine, the BOOTP table can be difficult to maintain on large networks. You may still encounter BOOTP in existing networks, but most likely it will support only diskless workstations (which are sometimes called network computers).

Dynamic Host Configuration Protocol (DHCP)

By now, you have seen several references to DHCP. **Dynamic Host Configuration Protocol (DHCP)** is an automated means of assigning a unique IP address to every device on a network. Reasons for implementing DHCP include the following:

- *To reduce the time and planning spent on IP address management*. Central management of IP addresses eliminates the need for network administrators to edit the TCP/IP configuration on every network workstation, printer, or other device.

- *To reduce the potential for errors in assigning IP addresses*. With DHCP, no possibility exists that a workstation will be assigned an invalid address, and almost no possibility exists that two workstations will attempt to use the same IP address and thereby cause network errors (occasionally the DHCP server software may make a mistake). On the other hand, when manually assigning IP addresses on each workstation, or even manually editing a BOOTP table, it is easy to type in the wrong address or use the same address twice.
- *To enable users to move their workstations and printers without having to change their TCP/IP configuration*. As long as a workstation is configured to obtain its IP address from a central server, the workstation can be attached anywhere on the network and receive a valid address.
- *To make IP addressing transparent for mobile users*. For example, if a salesperson brought her Windows 2000 laptop to your conference room to make an online presentation about Internet commerce, she could attach to your network and receive an IP address without having to change her laptop's configuration.

DHCP was developed by the Internet Engineering Task Force as a replacement for BOOTP. Unlike BOOTP, DHCP does not require the network administrator to maintain a table of IP and MAC addresses on the server. It does, however, require the network administrator in charge of IP address management to install and configure the DHCP service on a server (such as Windows NT, Windows 2000, NetWare 4.11 or higher, or UNIX) that can run DHCP.

DHCP Leasing Process

With DHCP, a device borrows, or **leases**, an IP address while it is attached to the network. In other words, it uses the IP address on a temporary basis. When, for example, a client logs off the network, it relinquishes the IP address, and the DHCP server can assign it to another device.

Configuring DHCP involves specifying a range of addresses that can be leased to any network device on a particular segment. As a network administrator, you configure the duration of the lease (in the configuration of the DHCP server) to be as short or long as necessary, from a matter of minutes to forever. Once the DHCP server is running, clients can attach to it and receive their unique IP addresses. More specifically, the client and server take the following steps to negotiate the client's first lease (this example applies to a workstation, but devices such as networked printers may also take advantage of DHCP):

1. When the client workstation starts (assuming it has the TCP/IP protocol installed and bound to the NIC), it sends out a DHCP discover packet in broadcast fashion via the UDP protocol to the DHCP/BOOTP server port (by default, port number 67).
2. Every DHCP server that is connected to the same subnet as the client receives the broadcast request. Each DHCP server responds with an available IP address, while simultaneously withholding that address from other clients.

The response message includes the available IP address, subnet mask, IP address of the DHCP server, and the lease duration. This message goes out through the DHCP/BOOTP port 68 in broadcast fashion. Because the client doesn't have an IP address, the DHCP server cannot send the information directly to the client.

In some instances, BOOTP and DHCP may appear lumped together under the same category or service. For example, if you are configuring a Hewlett-Packard LaserJet that uses a JetDirect print server card, you can select "BOOTP/DHCP" from the printer's TCP/IP Configuration menu. BOOTP and DHCP are not always distinguished as separate services because they appear the same to the client and use the same server ports to handle their communications to and from the server. The main difference between the two services lies in how the server software distributes IP addresses.

3. The client accepts the first IP address that it receives, responding with a broadcast message that essentially confirms to the DHCP that it wants to accept the address. Because this message is broadcast, all other DHCP servers that might have responded to the client's original query see this confirmation and hence return the IP addresses they had reserved for the client to their pool of available addresses.
4. When the selected DHCP server receives the confirmation, it replies—in a broadcast fashion—with an acknowledgment message. It also provides more information, such as DNS or gateway addresses that the client might have requested.

The preceding steps involve the exchange of only four packets and therefore do not usually increase the time it takes for a client to log onto the network. Figure 11-15 depicts the DHCP leasing process. The client and server do not have to repeat this exchange until the lease is terminated. The IP address will remain in the client's TCP/IP settings even after the device restarts.

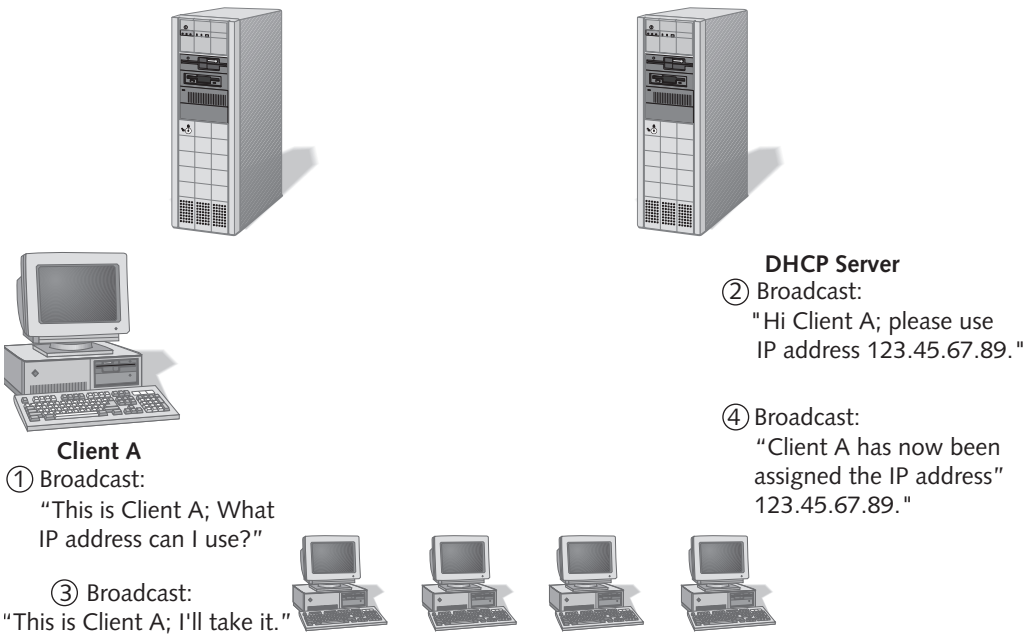

Figure 11-15 The DHCP leasing process

Terminating a DHCP Lease

A DHCP lease may expire based on the period established for it in the server configuration or it may be manually terminated at any time from either the client's TCP/IP configuration or the server's DHCP configuration. In some instances, a user must terminate a lease. Consider what might happen to the previously mentioned salesperson after she presents her online demonstration in your conference room. She returns to her office, plugs her laptop network cable into the outlet under her desk, and turns on her machine. Her TCP/IP settings will still contain the IP address and other information (such as DNS server and gateway address) she received from your DHCP server in the conference room. In addition, because the DHCP lease period lasts 30 days, her TCP/IP service will not attempt to pick up a new IP address from her own company's DHCP server. What will happen when the salesperson tries to pick up her e-mail? She will receive an error message, because her IP address will no longer be valid. Unfortunately, the error message will say only that it cannot establish a TCP/IP connection (not that a new IP address is needed). In this situation, the user needs to terminate her lease. In Windows terms, this event is called a **release** of the TCP/IP settings.

To release TCP/IP settings on a computer running the Windows 2000 operating system:

1. Click **Start**, point to **Programs**, point to **Accessories**, and then click **Command Prompt**. The Command Prompt window opens.
2. At the command prompt, type **`ipconfig /release`**.
3. Press **Enter**. The values for your IP Address, Subnet Mask, and Default Gateway in the IP Configuration dialog box revert to all zeros.

To release TCP/IP settings on a computer running the Windows 9x operating system:

1. Click **Start**, and then click **Run**. The Run dialog box opens.
2. Type **winipcfg**, and then click **OK**. The IP Configuration dialog box opens, displaying the workstation's TCP/IP settings.
3. To release the DHCP lease, click **Release All**.
4. The values for IP Address, Subnet Mask, and Default Gateway in the IP Configuration dialog box revert to all zeros. Click **OK** to close the IP Configuration dialog box.

Releasing old DHCP information is the first step in the process of obtaining a new IP address. In the preceding example, the salesperson would also have to instruct her TCP/IP service to request a new IP address. This task is easily accomplished from most workstations or laptops.

To obtain a new IP address on a Windows 2000 workstation:

1. Click **Start**, point to **Programs**, point to **Accessories**, and then click **Command Prompt**. The Command Prompt window opens.
2. At the command prompt, type **`ipconfig /renew`**.
3. Press **Enter**. The values for your IP Address and Subnet Mask will be appropriate for the subnet to which you are now attached.

To obtain a new IP address on a Windows 9x workstation:

1. Click **Start**, and then click **Run**. The Run dialog box opens.
2. Type **winipcfg**, and then click **OK**. The IP Configuration dialog box opens, displaying the workstation's TCP/IP settings.
3. Click **Renew All** to obtain new TCP/IP settings from the DHCP server.

   The values for your IP Address and Subnet Mask in the IP Configuration dialog box will be appropriate for the subnet to which you are now attached.
4. Click **OK** to close the IP Configuration dialog box.

With TCP/IP becoming the protocol of choice on most networks, you will most certainly have to work with DHCP—either from the client, the server side, or both—at some point in your networking career. As mentioned earlier, DHCP services run on several types of

servers. The installation and configurations for each type of server vary; for specifics, you must refer to the DHCP server software's manual. To qualify for Net+ certification, you need not know the intricacies of installing and configuring DHCP server software. You do, however, need to know what DHCP does and how it accomplishes it. You also need to understand the advantages of using DHCP rather than other means of assigning IP addresses.

Windows Internet Naming Service (WINS)

The **Windows Internet Naming Service (WINS)** provides a means of resolving NetBIOS names to IP addresses. Recall from Chapter 3 that NetBIOS is used primarily with Windows-based systems, and that a NetBIOS name is a unique alphanumeric name assigned to each Windows-based workstation on a network. WINS is used exclusively with systems that use NetBIOS—therefore, it usually appears on Windows-based systems. With fewer and fewer networks relying on NetBIOS, WINS is quickly becoming scarce.

A computer's NetBIOS name and its TCP/IP host name are different entities, though you can choose to use the same name for the NetBIOS name as you use for the TCP/IP name. Earlier, you learned that DNS provides resolutions of TCP/IP host names and IP addresses. WINS, on the other hand, provides resolution of NetBIOS names and IP addresses. Essentially, WINS has the same relationship to NetBIOS as DNS has to TCP/IP. That is, both WINS and DNS associate names with IP addresses.

Unlike DNS, however, WINS is an automated service that runs on a server. In this sense, it resembles DHCP. WINS may be implemented on servers running Windows NT Server version 3.5 or higher or on servers running Windows 2000 Server. It maintains a database on the server that accepts requests from Windows or DOS clients to register with a particular NetBIOS name. Note that WINS does not assign names or IP addresses, but merely keeps track of which NetBIOS names are linked to which IP addresses.

WINS offers several advantages:

- Guarantees that a unique NetBIOS name is used for each computer on a network. WINS manages which NetBIOS name is associated with each IP address, and it will not allow two machines with the same name to register.
- Support for DHCP. WINS can be integrated with the dynamic IP addressing method used by DHCP.
- Better network performance. As long as WINS manages the mappings between IP addresses and NetBIOS names, clients do not have to broadcast their NetBIOS names to the rest of the network. The elimination of this broadcast traffic improves network performance.

Every client workstation that needs to register with the WINS server must know how to find the server. Thus the WINS server cannot use a dynamic IP address (such as one assigned by a DHCP server). Instead, a specific IP address must be assigned to it manually.

To configure a Windows 2000 workstation to use the WINS service:

1. Click **Start**, point to **Settings**, and then click **Control Panel**.
2. Double-click **Network and Dial-up Connections**. The Network and Dial-up Connections window opens.
3. Right-click the **Local Area Connection** icon, and then click **Properties** in the shortcut menu. The Local Area Connection Properties dialog box appears.
4. Highlight **Internet Protocol (TCP/IP)** in the list of network components, and then click **Properties**. The Internet Protocol (TCP/IP) Properties dialog box opens.
5. Click the **Advanced** button. The Advanced TCP/IP Settings dialog box opens, as shown in Figure 11-16.

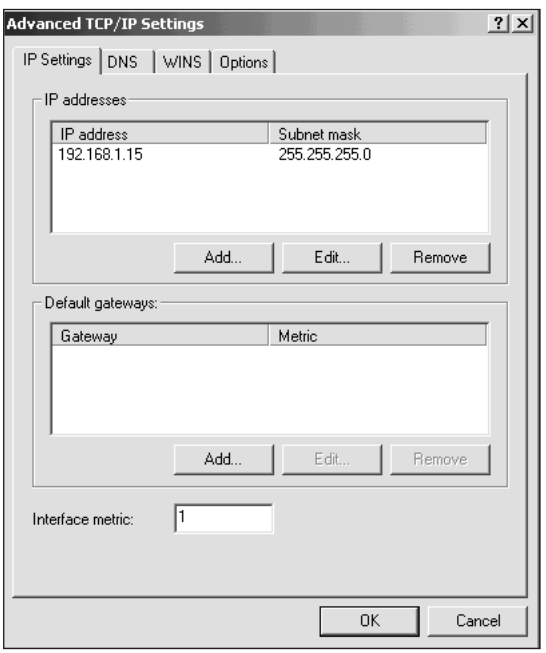

Figure 11-16 Advanced TCP/IP Settings dialog box in Windows 2000

6. Click the **WINS** tab.
7. To add a WINS server address, click **Add**. The TCP/IP WINS Server dialog box appears.
8. Enter the IP address of your WINS server in the WINS server search order text box. Click **Add** to add this server's address to your list of WINS servers.
9. If you have a secondary WINS server, you can add that by repeating Steps 7 and 8.

10. Click **OK** to close the Advanced TCP/IP Settings dialog box and save your changes.
11. Click **OK** to close the Internet Protocol (TCP/IP) Properties dialog box.
12. Click **OK** to close the Local Area Connection Properties dialog box.

To configure a Windows 9x workstation to use the WINS service:

1. Right-click the **Network Neighborhood** icon, and then click **Properties** in the shortcut menu. The Network dialog box opens.
2. Click the **Configuration** tab.
3. Select the TCP/IP protocol that is bound to your network adapter in the list of installed networking components, and then click **Properties**. The TCP/IP Properties dialog box opens.
4. Click the **WINS Configuration** tab.
5. To establish the identity of your WINS server, click **Enable WINS Resolution**.
6. Type the IP address of your WINS server in the spaces provided under "WINS Server Search Order" and then click **Add**.
7. If you have a secondary WINS server, repeat Step 6 to add its IP address number.
8. Click **OK** to save your changes.
9. Click **OK** to exit the Network dialog box.
10. You will be asked whether you want to restart your computer to save your changes. Click **Yes** to confirm that you want to restart it.

As with DNS, a complete discussion of WINS could fill at least an entire chapter. If you plan to specialize in Windows NT or Windows 2000 networking and clients running NetBIOS, you should investigate this topic further. For Net+ certification, you should be familiar with the purpose and advantages of WINS.

Addressing in IPv6

Up to this point, you have learned about IP addressing according to the IPv4 scheme. This section introduces you to addressing in IPv6 and the differences between addressing in IPv4 and addressing in IPv6.

As you have learned, IP version 6 (IPv6) (also known as **IP next generation**, or **IPng**) is slated to replace the current IP protocol, IPv4. Some applications, operating systems, and servers already provide support for IPv6, but many organizations have not made the switch due to the anticipated difficulty of changing their addressing scheme. In the coming years, however, switching to IPv6 will become not only desirable, but imperative. IPv6 offers several advantages over IPv4, including a more efficient header, the inclusion of the IPSec security technique, better support for QoS, and automatic IP address configuration.

But perhaps the most valuable advantage IPv6 offers is its promise of billions and billions of additional IP addresses through its new addressing scheme. As you have learned, the explosive growth of the Internet has meant that nearly all the traditional IP addresses are now in use. Scientists who developed the Internet over 20 years ago did not foresee this growth and therefore, did not make allowances for the volume of IP addresses the global network would require. IPv6 has been designed to correct this problem.

The most notable difference between IP addresses in IPv4 and IPv6 is their size. While IPv4 addresses are composed of four 8-bit fields (or octets), for a total of 32 bits, IPv6 addresses are composed of eight 16-bit fields and total 128 bits. The added fields and the larger address size result in an increase of 2^{96} (or 4 billion times 4 billion times 4 billion) available IP addresses in the IPv6 addressing scheme. The addition of more IP addresses not only allows every interface on every Internet-connected device to have a unique number, but also eliminates the need for IP address conservation. For example, organizations that use NAT through IP gateways in order to share limited IP addresses among several workstations will no longer need to do so after they adopt IPv6 addressing.

A second difference between IPv4 and IPv6 addresses is the way they are represented. While each octet in an IPv4 address contains binary numbers separated by a period (for example, 123.45.67.89), each field in an IPv6 address contains hexadecimal numbers separated by a colon. An example of a valid IPv6 address is F:F:0:0:0:0:3012:0CE3. Because many IPv6 addresses will contain multiple fields that have values of 0, a shorthand for representing these fields has been established. This shorthand substitutes "::" for any number of multiple, zero-value fields. Thus, the IPv6 address example above could be also be written as FF::3012:0CE3. An interesting, easily shortened address is the IPv6 loopback address. Recall that in IPv4 the loopback address has a value of 127.0.0.1. In IPv6, however, the loopback address has a value of 0:0:0:0:0:0:0:1. Abbreviated, the IPv6 loopback address becomes ::1. The substitution of multiple zero value fields can only be performed once within an address; otherwise, you would not be able to tell how many fields the "::" symbol represented.

A third difference between the two types of IP addresses is that in IPv6, each interface on a device is assigned its own IP address, rather than each node having its own IP address. Thus, if you were the network administrator for a network containing two VLANs, each associated with a different port on a switch, you could issue data transmissions to all interfaces on one VLAN by simply directing your transmission to the switch port associated with that VLAN. In addition to allowing different interfaces to have different IP addresses, IPv6 addressing distinguishes between different types of interfaces. One type of IPv6 address is a **unicast address**, or an address that represents a single interface on a device. A unicast address is the type of address that would be assigned, for example, to a workstation's NIC or a hub port. A **multicast address** represents multiple interfaces (often on multiple devices). Multicast addresses are useful for transmitting the same data to many different devices simultaneously. In IPv6, multicast addressing prevents the need for a broadcast address. Thus, there is no such thing as a broadcast address in IPv6. An **anycast address** represents any one interface from a group of interfaces (often on multiple nodes), any one of which (usually the first available) can accept a transmission.

Anycast addresses could be useful for identifying all of the routers that belong to one ISP, for example. In this instance, an Internet transmission destined for one of that ISP's servers could be accepted by the first available router in the anycast group. The result is that the transmission finishes faster than if it had to wait for one specific router interface to become available. At this time, anycast addresses are not designed to be assigned to hosts, such as servers or workstations.

A fourth significant difference between IPv4 and IPv6 addressing is that in IPv6, each address contains a **Format Prefix**, or a variable-length field at the beginning of the address that indicates what type of address it is. The Format Prefix also establishes the arrangement of the rest of the address's fields. In the IPv4 addressing scheme, no distinction is made between an address that represents one device or interface and an address that represents multiple devices or interfaces. For example, if you used the `netstat` command and noticed that your workstation was connected to a device with the IP address of 161.45.03.88, you could not conclude anything about the device or its transmission from that number. However, in IPv6, the first field of the IP address would provide a clue as to what type of interface the address represented. A unicast or anycast address begins with one of the two following hexadecimal strings: FEC0 or FE80. A multicast address begins with the following hexadecimal string: FF0*x*, where *x* is a character that corresponds to a group scope ID (for example, a group of addresses that belongs to an entire organization or a group of addresses that belongs to one site on a WAN).

Although IPv6 has been defined since the mid-1990s, organizations have been slow to adopt it. However, the use of IPv6 is predicted to grow rapidly as more and more devices (particularly wireless electronics) are connected to the Internet. During this transition phase, IPv4 and IPv6 will need to coexist. To do so, modern connectivity devices will most likely translate IPv4 addresses into IPv6 addresses for transmission over the Internet by padding the extra fields with zeros to fill the 128-bit address space.

TCP/IP Subprotocols

In Chapter 3, you learned that TCP/IP is not a single protocol, but rather a suite of protocols, commonly called subprotocols, each of which performs a distinct function. That chapter introduced the core subprotocols, including IP, TCP, UDP, ICMP, and ARP, as well as Application layer protocols such as Telnet, FTP, SMTP, and SNMP. This section briefly reviews these subprotocols and introduces several new subprotocols. In addition, it describes in more depth the subprotocols with finite purposes such as SMTP and POP, as opposed to the more general-purpose subprotocols such as TCP and UDP.

In your networking career, you will need to be familiar with all of the subprotocols covered in this book, even if you do not choose to master the fine points of TCP/IP networking. Suppose, for instance, that you are troubleshooting a problem with the e-mail package at your organization. Before you can talk to the vendor's technical support personnel, you must know whether your e-mail software uses POP or IMAP. In troubleshooting

and managing a network, you will encounter many situations such as this one that require you to know which TCP/IP subprotocols your network uses and how those subprotocols are implemented.

A Review of TCP/IP Subprotocols

The following list of subprotocols and their functions should look familiar to you. If you do not remember how they fit into the OSI Model or what some of the terms (such as "connectionless") mean, you should review the summary at the end of Chapter 3.

- *Internet Protocol (IP)*—A core protocol in the TCP/IP suite that belongs to the Internet layer of the TCP/IP model and provides information about how and where data should be delivered. IP is the subprotocol that enables TCP/IP to internetwork.
- *Transmission Control Protocol (TCP)*—A core protocol of the TCP/IP suite. TCP belongs to the Transport layer and provides reliable data delivery services because it is connection-oriented.
- *User Datagram Protocol (UDP)*—A core protocol in the TCP/IP suite that sits in the Transport layer, between the Internet layer and the Application layer of the TCP/IP model. Unlike TCP, UDP is a connectionless transport service.
- *Internet Control Message Protocol (ICMP)*—A core protocol in the TCP/IP suite that notifies the sender that something has gone wrong in the transmission process and that packets were not delivered.
- *Address Resolution Protocol (ARP)*—A core protocol in the TCP/IP suite that belongs in the Internet layer and obtains the MAC (physical) address of a host, or node, and then creates a local database that maps the MAC address to the host's IP (logical) address.
- *Telnet*—An Application layer terminal emulation protocol used to log onto remote hosts using TCP/IP.
- *File Transfer Protocol (FTP)*—An Application layer protocol used to send and receive files via TCP/IP.
- *Simple Network Management Protocol (SNMP)*—A communication protocol used to manage devices on a TCP/IP network.

Additional and Highlighted Subprotocols

In addition to the subprotocols introduced in Chapter 3, you should understand the subprotocols described in the following sections. Some of these will be new to you, while others (such as POP) will be familiar. Note that many of these protocols belong to the Application layer of the TCP/IP Model, which translates to the Application, Presentation, and Session layers of the OSI Model.

Reverse Address Resolution Protocol (RARP)

The Address Resolution Protocol (ARP) is a means of obtaining the MAC address of a local host and keeping that information in a local cache. If a device doesn't know its own IP address, however, it can't use ARP, because it cannot issue ARP requests or receive ARP replies. One solution to this problem is to allow the client to send a broadcast message with the MAC address of a device and receive the device's IP address in reply. This process, which is the reverse of ARP, is made possible by the **Reverse Address Resolution Protocol (RARP)**. A RARP server maintains a table of MAC addresses and their associated IP addresses (similar to a BOOTP table). By consulting this table, a RARP server can respond to a client's request for an IP address associated with a particular MAC address. Only a RARP server can provide this service. A network may use more than one RARP server to balance the load caused by RARP requests and responses.

RARP was originally developed as a means for diskless workstations to obtain IP addresses from a server before BOOTP emerged. Figure 11-17 illustrates how RARP can provide an IP address to a diskless workstation.

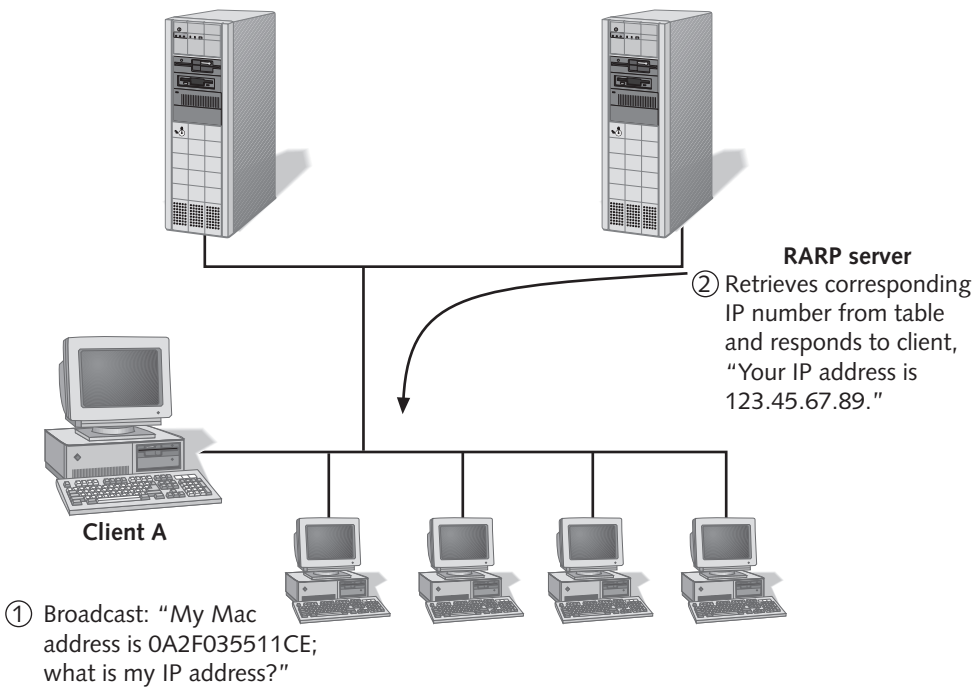

Figure 11-17 How RARP works

Simple Mail Transfer Protocol (SMTP)

In Chapter 3, you learned that the **Simple Mail Transfer Protocol (SMTP)** is responsible for moving messages from one e-mail server to another over TCP/IP-based networks. SMTP belongs to the Application layer of the TCP/IP Model and relies on TCP at the Transport layer. It operates from port 25. (That is, requests to receive mail and send mail go through port 25 on the SMTP server.) SMTP, which provides the basis for Internet e-mail service, relies on higher-level programs for its instructions. Although SMTP comes with a set of human-readable commands that you could conceivably use to transport mail from machine to machine, this method would be laborious, slow, and error-prone. Instead, other services, such as the UNIX sendmail software, provide more friendly and sophisticated mail interfaces that rely on SMTP as their means of transport.

SMTP is a simple subprotocol, incapable of doing anything more than transporting mail or holding it in a queue. In the post office analogy of data communications, SMTP is like the mail carrier who picks up his day's mail load at the post office and delivers it to the homes on his route. The mail carrier does not worry about where the mail is stored overnight or how it gets from another city's Post Office to his Post Office. If a piece of mail is undeliverable, he simply holds onto it; the mail carrier does not attempt to figure out what went wrong. In Internet e-mail transmission, higher-level mail protocols such as POP and IMAP, which are discussed in the next two sections, take care of these functions.

When you configure clients to use Internet e-mail, you need to identify the user's SMTP server. (Sometimes this server is called the mail server.) Each e-mail program will specify this setting in a different place, though most commonly in the Mail Preferences section. Assuming that your client uses DNS, you do not have to identify the IP address of the SMTP server-only the name. For example, if a user's e-mail address is jdoe@usmail.com, his SMTP server is probably called "usmail.com." You do not have to specify the TCP/IP port number used by SMTP, because both the client workstation and the server will assume that SMTP requests and responses flow through port 25.

Post Office Protocol (POP)

The **Post Office Protocol (POP)** is a protocol that provides centralized storage for e-mail messages. It belongs to the Application layer of the TCP/IP Model and relies on SMTP. In the Postal Service analogy, POP is like the Post Office that holds mail until its delivery to customers. A storage mechanism such as POP is necessary because users are not always logged onto the network and available for receiving messages. Both SMTP and a service such as POP are necessary for a mail server to receive, store, and forward messages. These two protocols cannot work without each other.

Users need an SMTP-compliant mail program to connect to their POP server and download mail from storage. POP does not allow users to keep the mail on the server after they retrieve it, which can create a problem for users who move from machine to machine. For example, if three receptionists share a company's front desk PC, each will

need a separate area on the hard disk to save mail if the network uses POP. But what happens if someone from the Accounting Department needs to fill in at the reception desk one afternoon, but also wants to read mail while stationed there. With POP, he would have to create yet another area for his mail on the front desk PC. When he returns to his desk in Accounting, however, the mail will not be accessible because it would have been saved on a different PC. A few options exist for circumventing this problem (such as keeping users' mail on a LAN server), but a more thorough solution has been provided by a new, more sophisticated e-mail protocol called IMAP, described in the next section.

The "POP" acronym has multiple meanings in the world of networking. In Chapter 7's discussion of remote connectivity, POP stood for a carrier's point of presence. In this chapter's discussion of Internet e-mail, POP stands for Post Office Protocol. Other acronyms also have double meanings in the computer world, so when reading or talking about data communications, you need to understand the particular context. Often the Post Office Protocol will be identified with its version number as well—for example, POP2 or POP3. To make matters more confusing, some networking professionals use POP as a verb, as in the following sentence: "I always pop for mail before I go to lunch."

Internet Mail Access Protocol (IMAP)

The **Internet Mail Access Protocol (IMAP)** is a mail storage and manipulation protocol that also depends on SMTP's transport system. IMAP was developed as a more sophisticated alternative to POP. The most current version of IMAP is version 4 (IMAP4). IMAP4 can (and eventually will) replace POP without the user having to change e-mail programs. The single biggest advantage IMAP4 has over POP is that users can store messages on the mail server, rather than always having to download them to a local machine. This feature benefits users who move from workstation to workstation. In addition, IMAP4 provides the following features:

- *Users can retrieve all or only a portion of any mail message.* The remainder can be left on the mail server. This feature benefits users who move from machine to machine and users who have slow connections to the network or minimal free hard disk space.
- *Users can review their messages and delete them while the messages remain on the server.* This feature preserves network bandwidth, especially when the messages are long or contain attached files, because the data need not travel over the wire from the server to the client's workstation. For users with a slow modem connection, deleting messages without having to download them represents a major advantage over POP.
- *Users can create sophisticated methods of organizing messages on the server.* A user might, for example, build a system of folders to contain messages with similar content. Also, a user might search through all of the messages for only those that contain one particular keyword or subject line.

- *Users can share a mailbox in a central location*. For example, if several maintenance personnel who use different PCs need to receive the same messages from the Facilities Department head but do not need e-mail for any other purpose, they can all log on with the same ID and share the same mailbox on the server. If POP were used in this situation, only one maintenance staff member could read the message; she would then have to forward or copy it to her colleagues.
- *IMAP4 can provide better security than POP because it supports authentication*. Security is an increasing concern for network managers as more organizations connect to the public Internet.

Although IMAP provides significant advantages over POP, it also comes with a few disadvantages. For instance, IMAP servers require more storage space and usually more processing resources than POP servers do. By extension, network managers must keep a closer watch on IMAP servers to ensure that users are not consuming more than their fair share of space on the server. In addition, if the IMAP server fails, users cannot access the mail left there. (IMAP does allow users to download messages to their own PCs, however.)

Until recently, another consideration was that most popular e-mail programs were designed for use with POP servers only. This standard is changing, however, and you should have no difficulty obtaining mail programs that use IMAP4. For example, Eudora Pro, GroupWise, Lotus Notes, Netscape, and Microsoft Outlook all support IMAP4.

Hypertext Transport Protocol (HTTP)

Hypertext Transport Protocol (HTTP) is a protocol that operates in the Application layer of the TCP/IP model. You can think of it as the language that Web clients and servers use to communicate. HTTP therefore forms the backbone of the Web. When you type the address of a Web page in your Web browser's address field, HTTP transports the information about your request to the Web server on port 80. It interprets your request and returns the Web server's information to you in **Hypertext Markup Language (HTML)**, the Web document formatting language. If you access a Web page that contains links to other Web pages, HTTP allows you to connect those links after you click on them. Figure 11-18 outlines this process.

HTTP/0.9, the original version of HTTP, was released in 1990. This version provided only the simplest means of transferring data over the Internet. Since then, HTTP has been greatly improved to make Web client/server connections more efficient, reliable, and secure. For example, HTTP/1.1, the current version of HTTP, allows servers to transmit multiple objects, such as text and graphics, over a single TCP connection using longer packets. It also allows a client to save Web pages via caching and to compare the saved pages with requested pages. If the two are identical, the Web browser will use the cached copy of the page to save bandwidth and time.

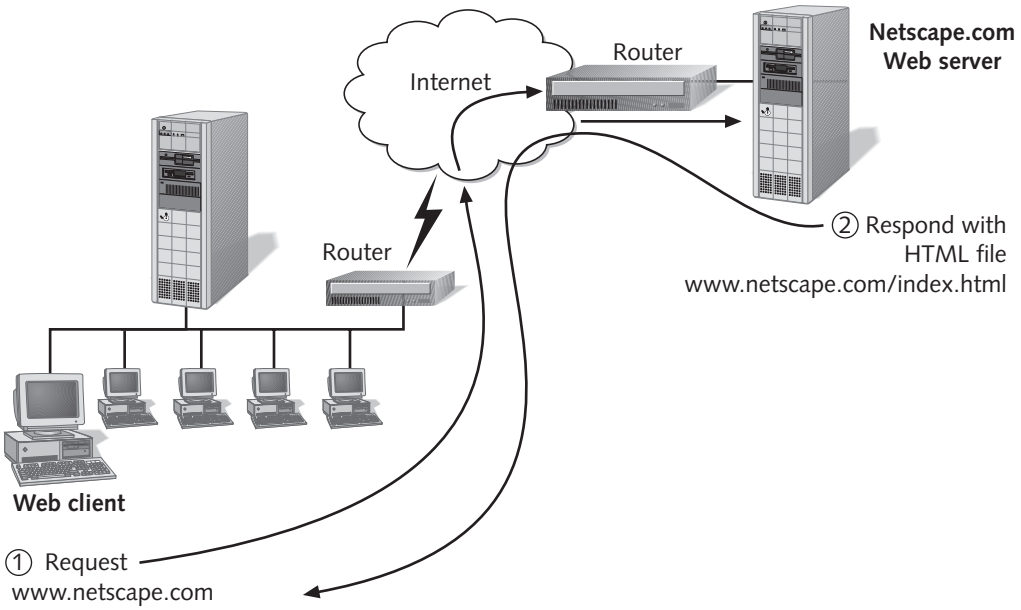

Figure 11-18 Web client/server transmission using HTTP

Network Time Protocol (NTP)

NTP, the **Network Time Protocol**, is used to synchronize the clocks of computers on a network. It is a very simple protocol that belongs to the Application layer of the TCP/IP Model and depends on UDP. Although it is simple, it is also important. Since many packets have a predefined period of time to reach their destination (after which they will be discarded), it is critical that all computer clocks on a network be synchronized. Otherwise, packets might expire prematurely, resulting in sporadic data loss. Time is also used in routing to determine the most efficient path for data over a network. NTP is a protocol that benefits from UDP's quick, connectionless nature at the Transport layer. NTP is time-sensitive and cannot wait for the error checking that TCP would require.

TCP/IP Troubleshooting

Of all network protocols, TCP/IP is the most likely to cause problems because it requires the most planning and post-installation configuration. As with any type of communication, many potential points of failure exist in the TCP/IP transmission process. Fortunately, TCP/IP comes with a complete set of troubleshooting tools that can help you to track down most TCP/IP-related problems without using expensive software or hardware to analyze network traffic. You should be familiar with the use of the following tools and their **switches** (the letters or words added to a command that allow you to customize the utility's output), not only because the Network+ certification exam

covers them, but also because you will regularly need these diagnostics in your work with TCP/IP networks. Each of these utilities can be accessed from the command prompt on a server or client running TCP/IP.

Each TCP/IP utility can be used by nearly any client that runs TCP/IP. However, the syntax of these commands may differ, depending on your client's operating system. For example, the command that traces the path of packets from one host to another is known as `traceroute` in the UNIX operating system, but as `tracert` in the Windows operating systems. Similarly, the options used with each command may differ according to the operating system. When working on a UNIX system, you can limit the maximum number of router hops the `traceroute` command allows by typing the `–m` switch. On a Windows system, the `–h` switch accomplishes the same thing. The following sections focus on the proper command syntax for Windows-based computers, but also refer to UNIX-specific commands when relevant.

ARP

You have learned that ARP is a protocol that obtains the MAC address of a host and then creates a local database that maps the MAC address to the host's IP address. The database that lists the associated MAC and IP addresses is called an **ARP table**. An ARP table contains two types of entries: dynamic and static. **Dynamic ARP table entries** are created when a client makes an ARP request that cannot be satisfied by data already in the ARP table. **Static ARP table entries** are those that a networking professional has entered using the ARP utility. The ARP utility provides a way of obtaining information from and manipulating a device's ARP table. It can be a valuable troubleshooting tool for discovering the identity of a machine whose IP address you know, or for solving the problem of two machines trying to use the same IP address.

Typing the `arp` command only from a Windows system will display the proper syntax and list of switches available for this command. To return useful data, the `arp` command requires at least one switch. For example, typing `arp –a` provides the entire ARP table for your host. Following is a list of the most popular ARP switches.

- `-a`—Displays the ARP table for the host at which you issue the command
- `-g`—Does the same thing as the `–a` switch
- `-d`—Removes an entry from the ARP table; this switch must be followed by the hostname corresponding to the ARP entry that you wish to remove
- `-s`—Adds an entry to the ARP table—in other words, creates a static ARP table entry. This switch must be followed by the host name and MAC address of the device you wish to add. On a server, using the `arp` command with this switch will only work if you are logged in as an administrator.
- *`hostname`*—You must replace "hostname" with a network host name. The command then lists the ARP table entry for the device with that host name.

Packet Internet Groper (PING)

The **Packet Internet Groper (PING)** is a utility that can verify that TCP/IP is installed, bound to the NIC, configured correctly, and communicating with the network. It is often employed simply to determine whether a host is responding (or "up"). PING uses ICMP to send echo request and echo reply messages that determine the validity of an IP address. These two types of messages work much in the same way that sonar operates. First, a signal, called an **echo request**, is sent out to another computer. The other computer then rebroadcasts the signal, in the form of an **echo reply**, to the sender. The process of sending this signal back and forth is known as **pinging**.

You can ping either an IP address or a host name. For example, to determine whether the ftp.netscape.com site is responding, you could type: `ping ftp.netscape.com` and press Enter. Alternately, you could type: `ping 205.188.212.121` (the IP address of this site) and press Enter. If the site is operating correctly, you would receive a response that includes multiple replies from that host. If the site is not operating correctly, you will receive a response indicating that the request timed out or that the host was not found. Figure 11-19 gives examples of a successful and an unsuccessful ping.

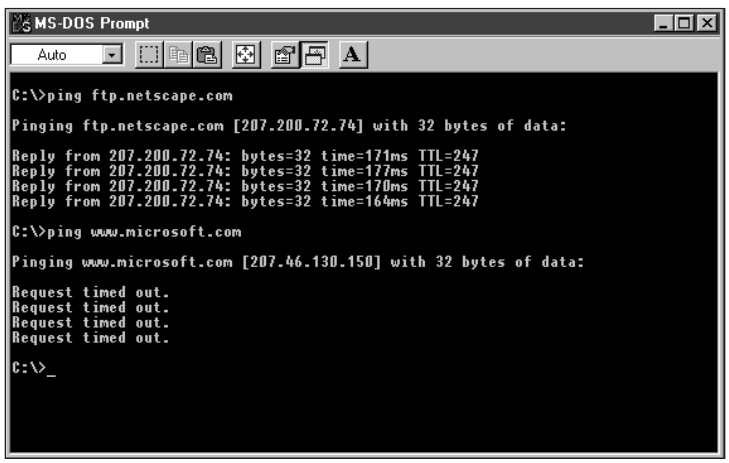

Figure 11-19 Example of a successful and an unsuccessful PING test

By pinging the loopback address, 127.0.0.1, you can determine whether your workstation's TCP/IP services are running. The loopback address automatically transmits a message back to the sending computer—that is, the message "loops back" to the sender. By pinging a host on another subnet, you can determine whether the problem lies with your gateway or DNS server.

For example, suppose that you have recently moved from the Accounting Department to the Advertising Department, and now you cannot access the Web. The first test you should perform is pinging the loopback address. If that test is successful, then you know that your workstation's TCP/IP services are running correctly. Next, you might try pinging your neighbor's

machine. If you receive a positive response, you know that your network connection is working. You should then try pinging a machine on another subnet that you know is connected to the network—for example, a computer in the IT department. If this test is unsuccessful, you can safely conclude that you do not have the correct gateway or DNS settings in your TCP/IP configuration or that your organization's gateway is malfunctioning.

As with other TCP/IP commands, PING can be used with a number of different options, or switches. A ping command begins with the word "ping" followed by a hyphen (-) and a switch, followed by a variable pertaining to that switch. For example, if you wanted to ping Netscape's FTP site with only two echo requests (rather than the standard four), you could type the following command: `ping –n 2 ftp.netscape.com`. The following list describes some of the most common switches used with the PING utility:

- `-?`—Displays the help text for the PING command, including its syntax and a full list of switches
- `-a`—When used with an IP address, resolves the address to a host name
- `-n`—Allows you to specify a number of echo requests to send
- `-r`—When used with a number from 1 to 9, displays the route taken during ping hops
- `-w`—Limits the time to wait for each echo response to a specific number of milliseconds (requires the specification of the number of milliseconds to wait)

Netstat

The **netstat** utility displays TCP/IP statistics and details about TCP/IP components and connections on a host. Information that can be obtained from the `netstat` command include: the port on which a particular TCP/IP service is running, whether or not a remote node is logged into a host, which network connections are currently established for a client, how many packets have been handled by a network interface since it was activated, and how many data errors have occurred on a particular network interface. As you can imagine, with so much information available, the netstat utility makes a powerful diagnostic tool.

For example, suppose you are a network administrator in charge of maintaining file, print, Web, and Internet servers for an organization. You discover that your Web server, which has multiple processors, sufficient hard disk space, and multiple NICs, is suddenly taking twice as long to respond to HTTP requests. Of course, you would want to check the server's memory resources as well as its Web server software to determine that nothing is wrong with either of those. In addition, you can use the netstat utility to determine the characteristics of the traffic going in and out of each network interface card. You may discover that one network card is consistently handling 80 percent of the traffic, even though you had configured the server to share traffic equally among the two. This fact may lead you to run hardware diagnostics on the NIC, and perhaps discover that its onboard processor has failed, making it much slower than the other NIC. Netstat

provides a quick way to view traffic statistics, without having to run a more complex program such as Windows 2000 Network Monitor.

If you use the **netstat** command without any switches, it will display a list of all the active TCP/IP connections on your machine, including the Transport layer protocol used (UDP or TCP), packets sent and received, IP address, and state of those connections, as shown in Figure 11-20.

```
% netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 172.16.1.1:139          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:113             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:631             0.0.0.0:*               LISTEN
tcp        0      0 172.16.1.1:139          172.16.1.128:1036       ESTABLISHED
tcp        0      0 10.1.1.10:33580         10.1.1.8:22             ESTABLISHED
tcp        0      0 10.1.1.10:33002         10.1.1.2:22             ESTABLISHED
udp        0      0 127.0.0.1:32776         0.0.0.0:*
udp        0      0 172.16.1.1:137          0.0.0.0:*
udp        0      0 0.0.0.0:137             0.0.0.0:*
udp        0      0 172.16.1.1:138          0.0.0.0:*
udp        0      0 0.0.0.0:138             0.0.0.0:*
udp        0      0 0.0.0.0:68              0.0.0.0:*
udp        0      0 0.0.0.0:68              0.0.0.0:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     2254   /tmp/.font-unix/fs-1
unix  11     [ ]         DGRAM                    1393   /dev/log
unix  2      [ ACC ]     STREAM     LISTENING     2949   /tmp/orbit-dklann/orb-1
434852031454701316
```

Figure 11-20 Output of a simple `netstat` command

11

However, as with other TCP/IP commands, netstat can be used with a number of different switches. A **netstat** command begins with the word **netstat** followed by a hyphen and a switch, followed by a variable pertaining to that switch, if required. For example, **netstat -a** displays all current TCP and UDP connections from the issuing device to other devices on the network, as well as the source and destination service ports. The **netstat -r** command allows you to post a listing of the routing table on a given machine. The following list describes some of the most common switches used with the netstat utility:

- **-a**—Provides a listing of all available TCP and UDP connections, even if they are simply listening and not currently exchanging data
- **-e**—Displays details about all the packets that have been sent over a network interface
- **-n**—Lists currently connected hosts according to their port and IP address (in numerical form)
- **-p**—Allows you to specify what type of protocol statistics to list; this switch must be followed by a protocol specification (TCP or UDP)
- **-R**—Provides a list of routing table information
- **-S**—Provides statistics about each packet transmitted by a host, separated according to protocol type (for example, IP, TCP, UDP, or ICMP)

Nbtstat

Recall from Chapter 3 that NetBIOS is a protocol that runs in the Session and Transport layers of the OSI Model and associates NetBIOS names with workstations. NetBIOS alone is not routable because it does not contain Network layer information. However, when encapsulated in another protocol such as TCP/IP, it can be routed. On networks that run NetBIOS over TCP/IP, the **nbtstat** utility can provide information about NetBIOS statistics and resolve NetBIOS names to their IP addresses. In other words, if you know the NetBIOS name of a workstation, you can use nbtstat to determine its IP address.

Nbtstat is useful on networks that run Windows-based operating systems and NetBIOS (because Novell and UNIX operating systems do not use NetBIOS, nbtstat is not useful on these types of networks). For example, suppose you are a network technician sitting at the help desk and troubleshooting a connectivity problem with a user's workstation. Suppose also that your network uses NetBIOS over TCP/IP. The user tells you she cannot view her files on the server. Although this may be a symptom of a permissions problem or a client software problem, you decide to check the obvious first. You quickly type the `nbtstat -all` and you discover that no other computer names appear. You conclude that her NetBIOS protocol is probably not functioning properly, is not installed, or is incorrectly configured, so you begin your troubleshooting by examining her workstation's network protocol properties.

As more and more networks run pure TCP/IP (and not NetBIOS over TCP/IP), nbtstat is becoming a less popular TCP/IP diagnostic utility.

As with netstat, nbtstat offers a variety of switches to tailor the output of the command. For example, you can type `nbtstat -A ip_address` to determine what machine is registered to a given IP address. Popular switches used with the nbtstat command are listed below. Notice that they are case-sensitive; the `–a` switch has a different meaning than the `–A` switch.

- `-a`—Displays a machine's name table given its NetBIOS name; the name of the machine must be supplied after the `–a` switch
- `-A`—Displays a machine's name table given its IP address; the IP address of the machine must be supplied after the `–A` switch
- `-n`—Displays the local machine's name table; it does not require a variable after the switch since it operates on the local machine
- `-r`—Lists statistics about names that have been resolved to IP addresses by broadcast and by WINS. This switch is useful to determine whether a workstation is resolving names properly or to see whether WINS is operating correctly.
- `-s`—Displays a list of all the current NetBIOS sessions for a machine. When the small letter `s` is used, the `nbtstat` command attempts to resolve IP

addresses to NetBIOS names in the listing. If the machine has no current NetBIOS connections, the result of this command will indicate that fact.

- `-S`—Displays a list of all the current NetBIOS sessions for a machine according to their IP address. If the machine has no current NetBIOS connections, the result of this command will indicate that fact.

Nslookup

The **nslookup** utility allows you to query the DNS database from any computer on the network. Using nslookup, you can find the DNS host name of a device by specifying its IP address, or vice versa. This ability is useful for verifying that a host is configured correctly or for troubleshooting DNS resolution problems. For example, if you wanted to find out whether the host whose name is ftp.netscape.com is operational, you could type: `nslookup ftp.netscape.com` and press Enter. The response would look something like the output in Figure 11-21.

```
% nslookup ftp.netscape.com
Server:  proxy1.mdsn1.wi.home.com
Address:  24.6.204.15

Non-authoritative answer:
Name:    ftp.netscape.com
Address:  64.12.168.249

%
```

Figure 11-21 Output of a simple `nslookup` command

Notice that the command provides not only the host's IP address, but also provides the primary DNS server name and address that holds the record for this name. To find the host name of a device whose IP address you knew, you would type: `nslookup` *`ip_address`* and press Enter. The response would include not only the host name for that device, but also its IP address and the IP address and host name of its primary DNS server.

Nslookup is available on all UNIX systems as well as on Windows 2000 systems. The `nslookup` command does not come with the Windows 9x operating systems, however.

`Nslookup` can reveal much more than just the IP address or host name of a device. If you type the command by itself and press Enter, you will enter the nslookup utility, and your command prompt will change to a `>`. You can then use a number of UNIX-based commands to find out more about the contents of the DNS database. For example, you could view a list of all the host name and IP address correlations on a particular DNS

server by typing `ls`. Or you could set the period to wait for a response to five seconds by typing `timeout=5` (the default is 10 seconds). Many other nslookup options exist, and as with other UNIX-based commands, you can find the complete list of them in the nslookup man pages. (You learned about man pages, a reference system built into UNIX, in Ch. 10.) To exit the nslookup utility and return to the normal command prompt, type `exit`.

Tracert (Traceroute)

The **`traceroute`** command (also known as **`tracert`** on Windows systems) uses ICMP to trace the path from one networked node to another, identifying all intermediate hops between the two nodes. This utility is useful for determining router or subnet connectivity problems.

To find the route, the traceroute utility transmits a series of UDP datagrams to a specified destination, using either the IP address or the host name to identify the destination. The first three datagrams that traceroute transmits have their TTL (time to live) set to 1. Because the TTL determines how many more network hops a datagram can make, datagrams with a TTL of 1 expire as they hit the first router. When they expire, they are returned to the source—in this case, the node that began the traceroute. In this way, traceroute obtains the identity of the first router. After it learns about the first router in the path, traceroute transmits a series of datagrams with a TTL of 2. The process continues for the next router in the path, and then the third, fourth, and so on, until the destination node is reached. Traceroute also returns the amount of time it took for the datagrams to reach each router in the path.

You can infer from traceroute's method and output that this utility can help diagnose network congestion or network failures. Traceroute is not foolproof, however. In fact, its results can be misleading, because traceroute cannot detect router configuration problems or detect whether a router uses different send and receive interfaces. In addition, routers may not decrement the TTL value correctly at each stop in the path. Therefore, traceroute is best used on a network with which you are already familiar. You can then use your judgment and experience to compare the actual test results with what you anticipate the results should be.

The simplest form of the `traceroute` command is `traceroute` ***`ip_address`***. On computers that use the Windows-based operating system, the proper syntax is `tracert` ***`ip_address`***. This command will return a list as shown in Figure 11-22.

```
% traceroute www.networksolutions.com
traceroute to www.networksolutions.com (216.168.224.69), 30 hops max, 38 byte packets
 1  * * *

% traceroute -I www.networksolutions.com
traceroute to www.networksolutions.com (216.168.224.69), 30 hops max, 38 byte packets
 1  * * *
 2  10.75.149.1 (10.75.149.1)  21.171 ms  21.026 ms  11.883 ms
 3  bb1-ge2-0.mdsn1.wi.home.net (24.6.204.1)  13.249 ms  14.739 ms  9.679 ms
 4  c2-se3-0-9.chcgil1.home.net (24.7.76.49)  13.511 ms  12.596 ms  18.576 ms
 5  aads.agis.net (206.220.243.19)  18.310 ms  26.789 ms  20.132 ms
 6  at-100100.inindrr01.us.telia.net (206.185.201.6)  68.421 ms  92.255 ms *
 7  at-0001.dcwdcrr01.us.telia.net (206.84.253.14)  97.508 ms  106.618 ms  100.920 ms
 8  ga011.herndon1.us.telia.net (206.84.235.249)  110.407 ms  108.272 ms  106.386 ms
 9  tii-internic.herndon1.us.telia.net (206.84.235.26)  96.980 ms  95.273 ms  92.744 ms
10  www.networksolutions.com (216.168.224.69)  93.718 ms  89.265 ms  88.828 ms
%
```

Figure 11-22 Output of a `traceroute` command

As with other TCP/IP commands, **`traceroute`** has a number of switches that may be used with the command. A **`traceroute`** command begins with either "**`traceroute`**" or "**`tracert`**" (depending on the operating system your computer uses), followed by a hyphen, a switch, followed by a variable pertaining to a particular switch, if required. For example, **`tracert -a`** displays all current TCP and UDP connections from the issuing device to other devices on the network, as well as the source and destination service ports. The following list describes some of the popular traceroute switches:

- **`-d`**—Instructs the **`tracert`** command not to resolve IP addresses to host names
- **`-h`**—Specifies the maximum number of hops the packets should take when attempting to reach a host (the default is 30); this switch must be followed by a variable
- **`-w`**—Identifies a timeout period for responses; this switch must be followed by a variable to indicate the number of milliseconds the utility should wait for a response

Ipconfig

Ipconfig is the TCP/IP administration utility for use with Windows NT and Windows 2000 operating systems. If you work with these operating systems, you will frequently use this tool to check a computer's TCP/IP configuration. It is a command-line based utility that provides information about a network adapter's IP address, subnet mask, and default gateway.

To use the ipconfig utility from a Windows 2000 computer, for example, click Start, point to Programs, point to Accessories, and then click Command Prompt to open the Command Prompt window. At the command prompt, type **`ipconfig`**. You should see TCP/IP information for your computer, similar to the output shown in Figure 11-23.

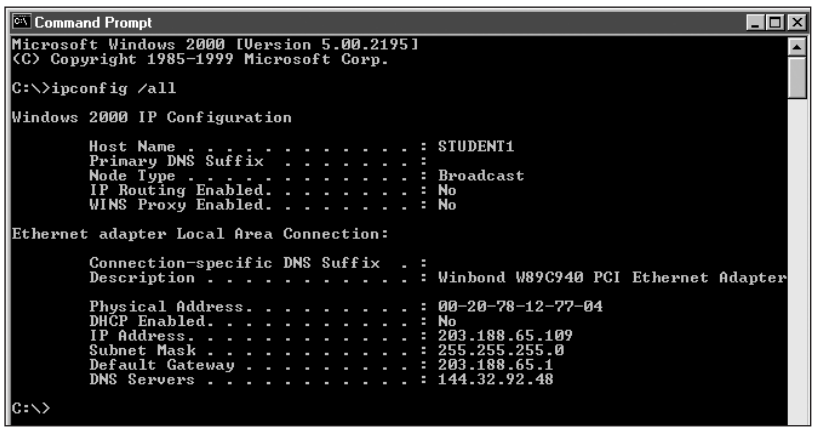

Figure 11-23 Output of an `ipconfig` command on a Windows 2000 workstation

In addition to being used alone to list information about the TCP/IP configuration, the ipconfig utility can be used with switches to manage a computer's TCP/IP settings. For example, if you wanted to view complete information about your TCP/IP settings, including your MAC address, when your DHCP lease expires, the address of your WINS server, and so on, you could type: `ipconfig /all`. Note that the syntax of this command differs slightly from other TCP/IP utilities. With `ipconfig`, a forward slash (/) precedes the command switches, rather than a hyphen. The following list describes some popular switches that can be used with the `ipconfig` command.

- `/?`—Displays a list of switches available for use with the `ipconfig` command
- `/all`—Displays complete TCP/IP configuration information for each network interface on that device
- `/release_all`—Releases DHCP-assigned addresses for all of the device's network interfaces
- `/renew_all`—Renews DHCP-assigned addresses for all of the device's network interfaces

Winipcfg

The **winipcfg** utility performs the same TCP/IP configuration and management as the ipconfig utility, but applies to Windows 9x and Me operating systems. It differs also in that it supplies the user with a graphical interface. As with ipconfig, networking technicians frequently use the `winipcfg` command when diagnosing TCP/IP problems.

To launch the winipcfg utility from a Windows 9x workstation, click Start, and then click Run to open the Run dialog box. In the Open text box, type `winipcfg`, and then click OK. The Winipcfg dialog box appears, displaying your network adapter's MAC and IP addresses, as well as your subnet mask and default gateway, as shown in Figure 11-24.

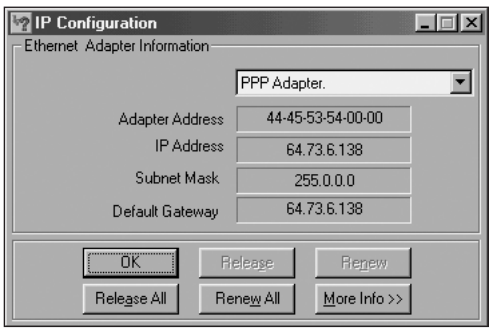

Figure 11-24 Winipcfg dialog box

As with the ipconfig utility, you can release or renew DHCP-assigned addresses through the Winipcfg dialog box. To renew all DHCP-assigned addresses, simply click the Renew All button. To release all DHCP-assigned addresses, simply click the Release All button. You also have the option to view more information about a machine's TCP/IP configuration. By clicking the More Info button, you can also view host name, node type, WINS server, when your DHCP lease was obtained, when it expires, and other information, as shown in Figure 11-25.

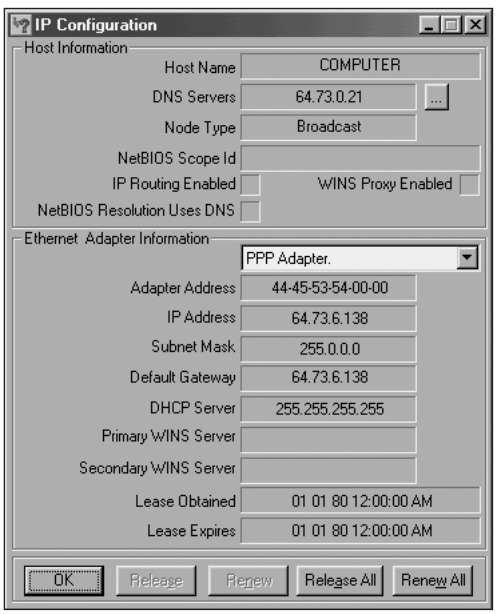

Figure 11-25 Detailed information available through winipcfg

Ifconfig

Ifconfig is the TCP/IP configuration and management utility used on UNIX systems. As with ipconfig on Windows 2000 systems and winipcfg on Windows 9x systems, ifconfig enables you to modify TCP/IP settings for a network interface, release and renew DHCP-assigned addresses, or simply check the status of your machine's TCP/IP settings. Ifconfig is also a utility that runs when a UNIX system starts, to establish the TCP/IP configuration for that system.

As with the other operating systems' TCP/IP configuration utilities, ifconfig can be used alone, or it can be used with switches to reveal more customized information. For example, if you wanted to view the TCP/IP information associated with every interface on a device, you could type: `ifconfig –a`. The output would resemble the output shown in Figure 11-26. Notice that the syntax of the `ifconfig` command uses a hyphen (-) before some of the switches and no preceding character for other switches. The following list describes some of the popular switches you may use with ifconfig. To view a complete list of options, you can read the ifconfig man pages. (You learned about man pages, a reference system built into UNIX, in Chapter 10.)

- `-a`—Applies the command to all interfaces on a device; can be used with other switches
- `auto-dhcp`—Automatically obtains an IP address from a DHCP server for an interface (as a shortcut, you can type simply "dhcp")
- `auto-dhcp release`—Releases the DHCP-assigned address from an interface
- `auto-dhcp status`—Displays the status of an interface's DHCP configuration
- `down`—Marks the interface as unavailable to the network
- `up`—Reinitializes the interface after it has been taken "down," so that it is once again available to the network

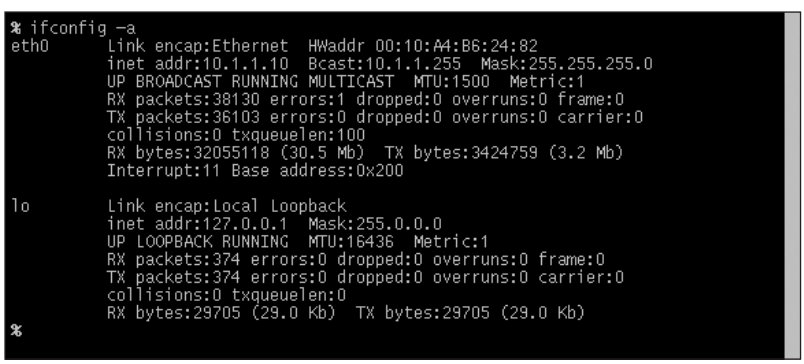

Figure 11-26 Detailed information available through ifconfig

APPLICATIONS AND SERVICES

By now, you should be familiar with some Internet services such as e-mail and FTP. The following sections discuss how TCP/IP networks provide these and other services, what protocols each service relies upon, and how these services benefit the network administrators who manage and support them. With the growth of Internet commerce and VPNs, chances are good that you will install, configure, or support at least one of the services described in the following sections.

World Wide Web (WWW)

In the most general sense, the **World Wide Web (WWW, or Web)** is a collection of internetworked servers that share resources and exchange information according to specific protocols and formats. On the client side, access to the Web requires the TCP/IP protocol, a unique IP address, a connection to the Internet, and a local interface to the Web called a **browser**. The two most popular browsers in use today are Netscape Navigator (or Communicator) and Microsoft's Internet Explorer. On the server side, a Web site requires the TCP/IP protocol, a connection to DNS servers, routers, Web server software, and remote connections to the Internet.

You have learned that Web servers and clients transmit content through HTTP and HTML. In addition, every Web page is identified by a **Uniform Resource Locator (URL)** that specifies the service it uses, its server's host name, and its HTML page or script name. For example, a valid URL is *http://www.gao.gov/fedrules.html*, where "*http*" is the service used by the page, "*www.gao.gov*" is the server's host name, and "*fedrules.html*" is the content page. If a URL does not specify the page name, the Web server displays a default page, called index.html (or index.htm) on most systems. On most new versions of browsers, you can type an unqualified host name instead of a fully qualified host name in the browser's URL field. An **unqualified host name** is a host name minus its prefix and suffix. For example, some browsers allow you to type simply "weather" in the URL field to retrieve the *www.weather.com* Web site. The browser would automatically add the prefix "*www*" and the suffix "*com*" to find the site. Not all URLs specify the HTTP protocol. They can also specify Telnet, file, or FTP, as in the following URL: *ftp://ftp.netscape.com/pub*.

As a network administrator, you may be charged with installing and configuring Web server software. You may have to choose from more than 100 Web server software types, and your decision must take into account your operating system, hardware, performance requirements, security requirements, budget, and existing software. Popular Web server software includes Apache, AOLServer, Lotus Domino, Microsoft's Internet Information Service, and Netscape's FastTrack. Although each program requires a different method of installation and configuration, all contain features for user account management, access (security) management, and content management.

As you can imagine, references for Web protocols, programming, and customization abound on the World Wide Web. If you're searching for more information about the topics covered in this chapter, you might want to start at a site that lists many different Web-related links, such as *www.webreference.com* or *www.internet.com*.

E-Mail

Through this chapter's discussion of SMTP, POP, and IMAP4, you have learned a great deal about how TCP/IP-based e-mail systems work. Currently, e-mail is the most frequently used and, therefore, the most relied-upon Internet service you will manage. Thus, you need to know how to support and troubleshoot your organization's e-mail package.

Although e-mail packages vary in how they look and, to some degree, how they act, they all work on the same principles. If a user cannot retrieve her e-mail messages from the server, you must verify her TCP/IP settings. If an entire department cannot retrieve or send e-mail, you should investigate possible problems with the department's gateway. Finally, if your entire organization's e-mail system fails, you must troubleshoot your mail server (or servers) and connection to the Internet.

Supporting and troubleshooting e-mail are no different than supporting and troubleshooting any other networked application. Because so many people depend on this service for their daily business, however, it's critical that you understand how it works.

File Transfer Protocol (FTP)

In Chapter 3, you learned that the **File Transfer Protocol (FTP)** manages file transfers between TCP/IP hosts. The FTP service depends on an FTP server that is always waiting for requests. Once a client connects to the FTP server, FTP data are exchanged via TCP using port 20. FTP commands are sent and received through TCP port 21. FTP belongs to the Application layer of the TCP/IP Model.

FTP is a simple, yet important part of the TCP/IP suite. Before the Web provided an easier means of transferring files, FTP commands were regularly used to exchange data between machines. FTP commands will still work without using browser software or special client software—that is, from the operating system's command prompt. As a network professional, you may need to use these commands to download software (such as NOS patches or client updates) from hosts. For example, if you need to pick up the latest version of the Novell Windows 2000 client, you can use FTP from your workstation's command prompt to download the compressed software from the ftp.novell.com server to your hard disk.

In order to do so, you must first start the FTP utility by typing `FTP` from the OS command prompt. The result is an FTP command prompt that appears as follows: `FTP>`. Once you have invoked the FTP utility, you can use the open command to connect to an FTP host. For example, to connect to Novell's FTP server, you would

type `open ftp.novell.com` at the FTP prompt. However, these two operations—starting FTP and connecting to a host—can be accomplished more simply through a single FTP command. For example, to FTP to Novell's FTP server from an OS prompt, you could type `FTP ftp.novell.com`, then press Enter to make the connection.

If the host is running, it will respond with a greeting and a request for you to log on, as shown in Figure 11-27. Many FTP hosts, especially those whose purpose is to provide software updates, accept anonymous logins. This means that when prompted for a user name, you need only type the word `anonymous` (in all small letters). When prompted for a password on an anonymous FTP site, you can usually use your e-mail address. The host's login screen should indicate whether this is acceptable. On the other hand, if you are logging onto a private FTP site, you must obtain a valid user name and password from the site's network administrator in order to make a successful connection.

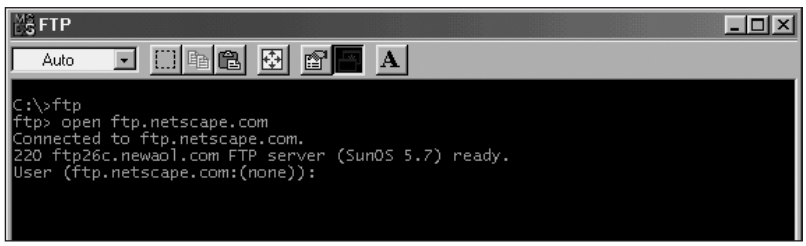

Figure 11-27 FTP login screen

Once you have successfully connected to a host, additional commands allow you to manage the connection and manipulate files. For example, after you have connected to Novell's FTP site, you could type: `cd public` to change your working directory to the "public" directory. Once in that directory, you could download a file by typing: `get XXX`, where "*XXX*" is the name of the file you want to download. To terminate the connection, simply type `quit`.

The following list summarizes a handful of useful FTP commands and their syntax. To learn more about these and other FTP commands, type `help` after starting the FTP utility.

- `ascii`—Sets the file transfer mode to "ASCII." Most FTP hosts store two types of files: ASCII and binary. Text files are typically ASCII-based and contain formatting characters, such as carriage returns. Binary files (for example, executable programs) typically contain no formatting characters. Before downloading files from an FTP host, you must understand what type of file you are downloading. If you download a file while in the wrong mode (ASCII if the file is binary or vice versa), your file will appear as gibberish when you open it. If the file you want to download is an ASCII file, type `ascii` at the FTP prompt and press Enter before starting your file transfer.

- `binary`—Sets the file transfer mode to "binary." If the file you want to download from an FTP site is binary (for example, an executable program or a compressed software patch) type `binary` at the FTP prompt and press Enter before starting your file transfer.
- `cd`—Changes your working directory on the host machine
- `delete`—Deletes a file on the host machine (provided you have permissions to do so)
- `get`—Transfers a file from the host machine to the client. For example, to transfer the file called update.exe from the host to your workstation, you can type: `get update.exe`. Unless you specify a target directory and filename, the file will be saved to your hard disk in the directory from where you started the FTP utility. Therefore, if you wanted to save the update.exe file to your C:\download\patches directory, you would type:

  ```
  get update.exe "c:\download\patches"
  ```

  (Make sure to include the quotation marks.)
- `help`—Provides a list of commands when issued from the FTP prompt. When used in conjunction with a command, it provides information on the purpose of that command. For example, after typing `help ls` you would learn that the `ls` command lists the contents of a remote directory.

You can substitute a "?" for the help command in order to learn more about FTP commands. For example, instead of typing: `help get`, you could type `? get` to learn more about the get command.

- `ls`—Lists the contents of the directory on the host where you are currently located.
- `mget`—Transfers multiple files from the FTP site to your workstation simultaneously. For example, to transfer all the text files within one directory, you could type: `mget *.txt` at the FTP> prompt.
- `mkdir`—Creates a new directory on the FTP host (provided you have permission to do so)
- `mput`—Transfers multiple files from your workstation to the FTP host
- `open`—Creates a connection with an FTP host
- `put`—Transfers a file from your workstation to the FTP host
- `quit`—Terminates your FTP connection and closes the FTP utility

As mentioned earlier, the advent of browsers and FTP client software have rendered this command-line method of FTPing files less common. What's more, modern FTP programs provide graphical interfaces for transferring files from a server to a client. Examples of popular, inexpensive (if not free) FTP clients include MacFTP, WS_FTP, CuteFTP, and SmartFTP. You can also accomplish FTP file transfers directly from a modern Web browser such as Internet Explorer or Netscape Communicator. In order to do this, you need only connect to the FTP host. From there, you can move through directories and exchange files just as you would on your desktop OS.

FTP and Telnet share some similarities, including their reliance on TCP and their ability to log onto a remote host and perform commands on that host. However, they differ in that, when you use Telnet, the commands you type require a syntax that is relative to your local workstation. When you use FTP, the commands you type require a syntax that is relative to the remote host that you have logged into. Also, Telnet has no built-in commands for transferring files between the remote host and your workstation.

Trivial File Transfer Protocol (TFTP)

The **Trivial File Transfer Protocol (TFTP)** is similar to FTP, in that it is a TCP/IP Application layer protocol that enables file transfers between computers. But TFTP relies on UDP at the Transport layer. Its use of UDP means that TFTP is connectionless and does not guarantee reliable delivery of data. Also, TFTP differs from FTP in that it does not actually log onto the remote host before enabling file transfers. Instead, when using TFTP, your computer issues a read request or a write request to the remote host. The remote host responds with an acknowledgement, then the two computers begin transferring data. Each time a packet of data is transmitted to the host, the local workstation waits for an acknowledgement from the host before issuing another packet. In this way, TFTP overcomes some of the limitations of relying on a connectionless Transport layer protocol.

TFTP is useful when you need to load programs on a diskless workstation. For example, suppose you ran a computer lab for a daycare facility that could only afford old computers with no hard disks but with 64 MB of memory. You could configure a server to hold multiple programs and respond to TFTP requests to transfer those programs into the diskless workstations' memory only when they are needed. In this situation, the fact that TFTP does not require a user to log onto a host is an advantage. It makes the transfer of program files quick and easy. (As you can imagine, however, in other cases, not requiring a login presents a security risk.)

In order for a file to be transferred via TFTP, it must be a file with full read and write privileges for all remote users. In other words, if you stored budget spreadsheets on your server and wanted to allow only the executives at your organization to download them, then TFTP could not be used as the means for accomplishing the downloads. Thus, before relying on TFTP you should make sure that the files to be transferred are not sensitive and are available to everyone connected to the network.

Gopher

Another Internet service that predates the WWW is **gopher**. A text-based utility, gopher allows you to navigate through a series of menus to find and read specific files. (The program is called "gopher" because it was developed at the University of Minnesota, whose mascot is the gopher.) Gopher is not sophisticated enough to interpret document formatting commands, such as HTML, but it does allow you to transfer files from one host to another by connecting with FTP. In addition, gopher was the first Internet interface to provide links from one host to another that are transparent to the user.

This utility requires a local gopher client and a gopher server. In the early 1990s, thousands of gopher servers provided information over the Internet. Gopher is rarely used today, however, because Web servers and browsers have made it obsolete.

Newsgroups

Newsgroups are similar to e-mail, in that they provide a means of conveying messages; they differ from e-mail in that they are distributed to a wide group of users at once rather than from one user to another. Newsgroups have been formed to discuss every conceivable topic, such as political issues, professional affiliations, entertainment interests, and sports. To belong to a newsgroup, a user subscribes to the server that hosts the newsgroup. From that point forward, the user receives all messages that other newsgroup members post with the newsgroup list as their mail-to address.

Newsgroups require news servers and, on the client side, e-mail programs capable of reading newsgroups or special newsgroup reading software. Rather than using SMTP, as e-mail does, newsgroup messages are transported by the **Network News Transport Protocol (NNTP)**. NNTP supports the process of reading newsgroup messages, posting new messages, and transferring news files between news servers. News servers are organized hierarchically, similarly to DNS servers. Your Internet service provider, for example, has a news server that uses a larger carrier's news server to communicate with other large carrier news servers.

E-commerce

One of the fastest growing sectors of the Internet is electronic commerce, or e-commerce. The term **e-commerce** refers to a means of conducting business over the Web—be it in retailing, banking, stock trading, consulting, or training. Any buying and selling of products or services that occurs over the Internet belongs in the e-commerce category. The first industries to take advantage of e-commerce were retailing and finance. In the past five years, more businesses have realized that e-commerce is critical to their success. Indeed, the number of online purchases has grown exponentially each year since 1996. You have probably used the Web to purchase books, music, clothes, or even furniture.

If you have an interest in Internet technologies, you may want to consider specializing in e-commerce. E-commerce involves customized HTML scripting, software programming,

multimedia, graphics, networking, and security skills. Because it often relies on credit card purchases or money transfers over the Internet, security is a significant concern. Web security is becoming more sophisticated to counter hackers, who continually find new ways to break into systems. Personal identification numbers and file encryption, for example, can no longer guarantee that information cannot be picked up by others in transit. Some day, we may use retinal patterns or fingerprints to provide secure access to Web sites. Chapter 15 discusses network security in more detail.

Voice over IP (VoIP)

Another growing service is **Voice over IP** (**VoIP**—pronounced "voyp"), the provision of telephone service over a TCP/IP network. When VoIP is carried over by Internet, it is often called **Internet telephony**. But not all VoIP calls are carried over the Internet. In fact, VoIP over private lines is a very effective and economical method of completing calls between two locations within an organization. And because the line is private, its congestion can be easily controlled, thus resulting in better sound quality than the Internet can provide. But given the Internet's breadth and low cost, it is appealing to consider the Internet for carrying conversations that we currently transmit over the PSTN.

Voice can be carried over TCP/IP networks in a variety of configurations. The following list describes three categories of VoIP technology:

- *Phone-to-phone*—In this configuration, two traditional telephones are connected through a TCP/IP network. On one end, a user picks up his telephone to make a call. His telephone is connected to a local telephone switch, which handles call routing for his business. The telephone switch accepts his voice signals, then passes them on to a gateway. Recall that a gateway is a combination of software and hardware that connects two dissimilar networks. In this case, the gateway connects the PSTN with a TCP/IP network (such as the Internet). The gateway may be located at a client's office or at a telephone carrier's facility. The gateway digitizes the analog voice signals, compresses the data, then assembles them into packets. As you have learned, packets contain routing and error-checking information, as well as data. The packets traverse the network and are accepted by another gateway at the receiving end. The receiving gateway reverses what the transmitting gateway did; that is, it disassembles the packets, decompresses the data, and converts it into an analog signal. The result is the original voice signal, which is passed to another telephone switch, to which the other telephone is connected. Both gateways perform their functions simultaneously. This enables the VoIP call to be full-duplex, just as a phone call over the PSTN, which means that both parties can speak and listen at the same time. Figure 11-28 depicts the configuration required for this type of VoIP call.

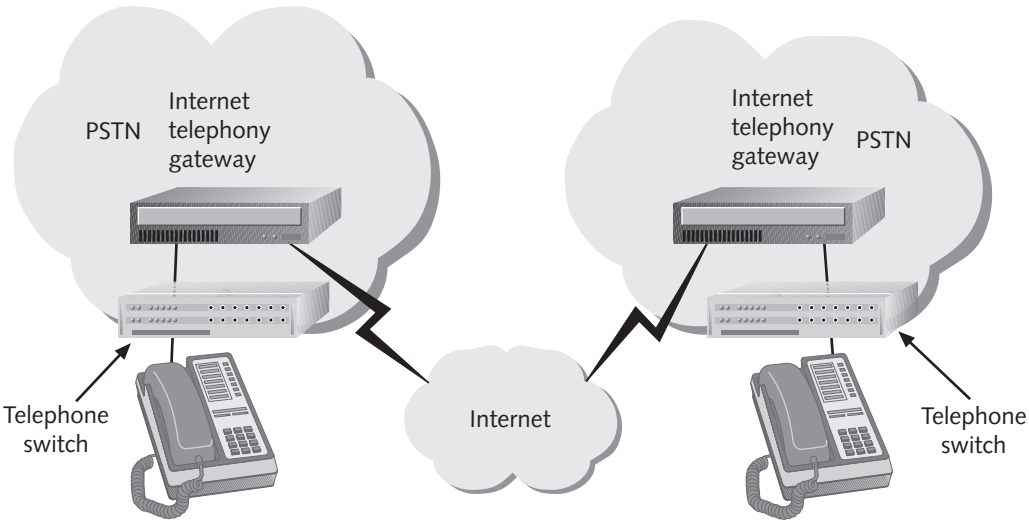

Figure 11-28 A phone-to-phone VoIP call

- *PC-to-phone*—In this scheme, one end of the call uses a PC, while the other end relies on a traditional telephone. On one end, a PC equipped with a microphone, speaker, and the appropriate software initiates calls over its network connection. Voice signals are converted to digital data by the PC's software, then transmitted through the PC's NIC and over the network just as any other data are transmitted. At the receiving end, a gateway accepts the data and translates them into voice signals (just as the receiving gateway does in the phone-to-phone VoIP call). These voice signals are then transmitted to a telephone switch, to which a traditional telephone is connected. (The process is reversed in a phone-to-PC configuration, when the traditional telephone user responds to the PC caller.) Figure 11-29 depicts this type of VoIP call.
- *PC-to-PC*—In this configuration, two PCs connect through a TCP/IP network to complete calls. The call is initiated just as a PC-to-phone call. On one end, a PC equipped with a microphone, speaker, and the appropriate software transforms voice signals into digital data. These data are transmitted over the network. On the other end, another PC with the same basic setup accepts the call from the network. During the call, users speak into the PC's microphone and listen via the PC's speakers. In this scheme, an IP address essentially becomes a telephone number, and the computer acts as a telephone. Figure 11-30 depicts this type of VoIP call.

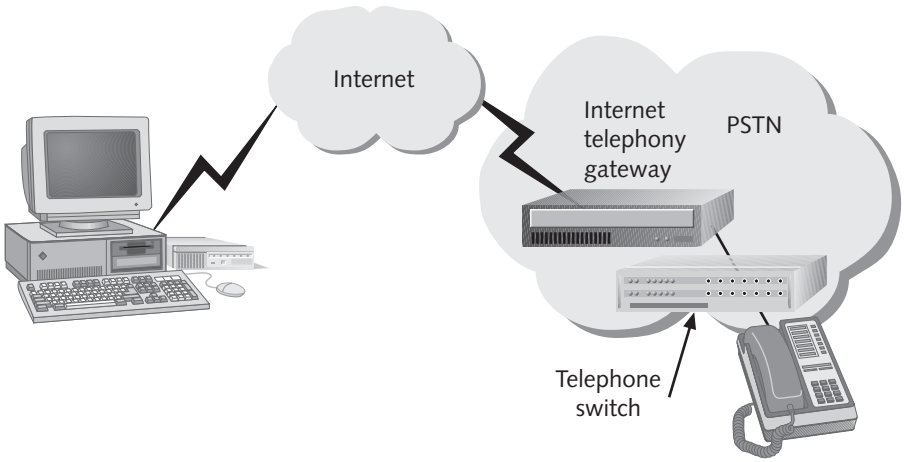

Figure 11-29 A PC-to-phone VoIP call

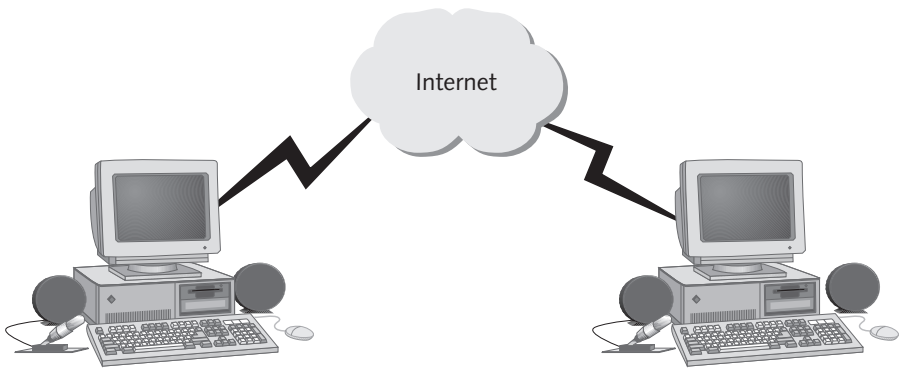

Figure 11-30 A PC-to-PC VoIP call

A tremendous benefit of VoIP, and Internet telephony in particular, is its low cost. For example, the cost of a call from New York to Tokyo would be the same as a call across town. Nevertheless, significant technical obstacles have prevented Internet telephony from becoming a widespread reality to date. First, more so than data transmissions, voice conversations can easily be distorted by the wire's quality of service. When you talk with your mother, you need to hear her syllables in the order in which she mouthed them, and preferably, without delay. (In contrast, data do not necessarily need to be received in the same order in which they were transmitted, because the destination node will sort the information out when it arrives.) Also, voice transmissions are subject to distortion if the connection becomes too noisy. In general, to prevent delays, disorder, and distortion, a voice connection requires more dedicated bandwidth than a data connection.

Another challenge to VoIP is the difficulty in billing for network-based calls, because their costs depend on so many factors. In addition, multiple carriers are typically

involved in any Internet-based call. Also, while standardization of Internet telephony and VoIP has begun, it is not even close to completion. The U.S. government is taking a hands-off approach in its regulation, letting the vendors and professional organizations sort out VoIP implementation. Therefore, VoIP is a rapidly changing area of networking. You can count on it to undergo significant modifications and continually improve in the coming years.

The use of networks to carry data, plus video and voice signals is known as **convergence**.

CHAPTER SUMMARY

- Every device on a TCP/IP-based network must have a unique IP address to ensure reliable data delivery. Without the correct IP address, data cannot be routed between networks and devices.
- Each IP address is a unique 32-bit number, divided into four groups of octets that are themselves separated by periods. An example of a valid IP address is 144.92.43.178. An IP address is typically represented in dotted decimal notation. It contains two types of information: network and host. It may also include subnet information.
- All nodes on a Class A network share the first octet of their IP numbers, a number between 1 and 126. Nodes on a Class B network share the first two octets, and all of their IP addresses begin with a number between 128 and 191. Class C network IP numbers share the first three octets, with their first octet being a number between 192 and 223.
- In addition to Class A, B, and C networks, Class D and Class E networks exist, although consumers and companies do not use them. Class D addresses begin with an octet whose value is between 224 and 239 and are reserved for a special type of transmission called multicasting. Class E addresses begin with an octet whose value is between 240 and 254, and are reserved for experimental use by IETF. Multicasting allows one device to send data to a specific group of devices (not the entire network segment) in a point-to-multipoint fashion.
- To use IP addresses more efficiently, the concept of subnetting was applied to the Internet in the mid-1980s. Subnetting is the process of subdividing a single class of network into multiple, smaller networks.
- Subnetting adds a third type of octet to the standard IP address. Rather than consisting of simply network and host information, a subnetted address includes network, subnet, and host information.
- The combination of an address's network and subnet information is called its extended network prefix. By interpreting an address's extended network prefix, a device can determine the subnet to which an address belongs.

- To determine whether an address is part of a subnet in the first place, a device interprets a subnet mask. A subnet mask is a special 32-bit number that, combined with a device's IP address, tells the rest of the network which kind of subnet the device is on. The bits of the subnet mask are set to 1 if the IP address information in the corresponding octets belongs to the extended network prefix. Otherwise, the subnet mask bits are 0, and the corresponding octets are assumed to represent host information.
- Routers external to an organization use only the network portion of the IP address to direct data to devices within that organization. External routers (such as those on the Internet) do not recognize subnets on specific LANs.
- Gateways are a combination of software and hardware that enable two different network segments to exchange data. In the context of IP addressing, a gateway facilitates communication between different subnets. Because one device on the network cannot send data directly to a device on another subnet, a gateway must intercede and hand off the information.
- Internet gateways maintain default routes to known addresses to expedite data transfer. The gateways that make up the Internet backbone are called core gateways. The Internet Network Operations Center (INOC) operates core gateways.
- A socket is a logical address assigned to a specific process running on a host computer. It forms a virtual connection between the host and client. The socket's address represents a combination of the host computer's IP address and the port number associated with a process.
- The use of port numbers simplifies TCP/IP communications and ensures that data are transmitted to the correct application. When a client requests communication with a server and specifies port 23, the server knows immediately that the client wants a Telnet session. No extra data exchange is necessary to define the session type, and the server can initiate the Telnet service without delay.
- Every host belongs to a domain; every domain is identified by its domain name. Usually, a domain name is associated with a company or other type of organization, such as a university or military unit.
- In the mid-1980s, the Network Information Center (NIC) at Stanford Research Institute devised a hierarchical way of tracking domain names and their addresses, called the Domain Name System (DNS). The DNS database does not rely on one file or even one server, but rather is distributed over several key computers across the Internet to prevent catastrophic failure if one or a few computers go down.
- Resolvers are any hosts on the Internet that need to look up domain name information. The resolver client is built into TCP/IP applications such as Telnet, HTTP, and FTP.
- Name servers are servers that contain databases of names and their associated IP addresses. A name server supplies a resolver with the requested information. If the name server cannot resolve the IP address, the query passes to a higher-level name server. Each name server manages a group of machines called a zone. DNS relies on the hierarchical zones to distribute naming information.

- When a host needs to communicate with other hosts on the Internet, that host must first find its name server. Large organizations often maintain more than one name server—a primary and a secondary name server—to help ensure Internet connectivity. When configuring the TCP/IP properties of a workstation, you need to specify a name server's IP address so that the workstation will know which machine to query when it must look up a name.
- A root server is a name server maintained by ICANN that is an authority on how to contact top-level domains, such as those ending with .com, .edu, .net, .us, and so on.
- To communicate with other devices through TCP/IP, every workstation, printer, or other node on a network requires a unique IP address. On the earliest TCP/IP networks, each device was manually assigned its own number through a local configuration file, and the number did not change until someone edited the configuration file.
- To ease IP address management, a service called the Bootstrap Protocol (BOOTP) was developed in the mid-1980s. BOOTP uses a central list of IP addresses and their associated devices' MAC addresses to assign IP addresses to clients.
- With BOOTP, a client does not have to remember its own IP address, and therefore network administrators do not have to manage each workstation on a network separately. This situation is ideal for diskless workstations.
- DHCP is an automated means of assigning a unique IP address to every device on a network. Reasons for implementing DHCP include the following: to reduce the time and planning spent on IP address management; to reduce the potential for human errors in assigning IP addresses; to enable users to move their workstations and printers without changing their TCP/IP configuration; and to make IP addressing transparent for mobile users.
- DHCP was developed by the IETF as a replacement for BOOTP. Unlike BOOTP, DHCP does not require the network administrator to maintain a table of IP and MAC addresses on the server. It does, however, require the network administrator in charge of IP address management to install and configure the DHCP service on a server.
- DHCP configuration involves specifying a range of addresses that can be leased to any network device on a particular segment. The term lease identifies how a client borrows a DHCP-assigned IP address.
- The Windows Internet Naming Service (WINS) provides a means of resolving NetBIOS names with IP addresses. WINS is used exclusively with systems that rely on NetBIOS—therefore, it is usually used on Windows-based systems.
- A computer's NetBIOS name and its TCP/IP host name are different entities, which may or may not be equivalent. DNS provides resolutions of host names and IP addresses, so you can think of WINS being to NetBIOS what DNS is to TCP/IP.
- The Address Resolution Protocol (ARP) is a means of obtaining the MAC address of a local host and keeping that information in a local cache. If a device doesn't

know its own IP address, however, it can't use ARP. A solution to this problem is to allow the client to send a broadcast message with the MAC address of a device and then receive the device's IP address in reply. This process, which is the reverse of ARP, is called the Reverse Address Resolution Protocol (RARP).

- Simple Mail Transfer Protocol (SMTP) is responsible for moving messages from one e-mail server to another over TCP/IP-based networks. SMTP operates through port 25, with requests to receive mail and send mail going through that port on the SMTP server.
- The Post Office Protocol (POP) runs on top of SMTP and provides centralized storage for e-mail messages. A storage mechanism such as POP is necessary because users are not always logged onto the network and available for receiving messages. Both SMTP and a service such as POP are required before a mail server can receive, store, and forward messages.
- The Internet Mail Access Protocol (IMAP), a mail storage and manipulation protocol that depends on SMTP's transport system, is a more sophisticated alternative to POP. The single biggest advantage IMAP4 has relative to POP is that it allows users to store messages on the mail server, rather than always having to download them to the local machine.
- Hypertext Transport Protocol (HTTP) is the language used by Web clients and servers to communicate with each other. When you type the address of a Web page in your Web browser's address field, HTTP transports the information about your request to the Web server and returns the Web server's information to you in the Hypertext Markup Language (HTML), the Web document formatting language.
- The Packet Internet Groper (PING) can verify that TCP/IP is installed, bound to the NIC, configured correctly, and communicating with the network. PING uses ICMP to send echo request and echo reply messages that determine the validity of an IP address.
- The netstat utility displays TCP/IP statistics and the state of current TCP/IP components and connections. It also displays ports, which can signal whether services are using the correct ports.
- The nbtstat utility provides information about NetBIOS names and their addresses. If you know the NetBIOS name of a workstation, you can use nbtstat to determine the workstation's IP address.
- The nslookup utility allows you to look up the DNS host name of a network node by specifying it's the node's IP address, or vice versa. This allows you to verify that a host is configured correctly. Nslookup is also useful for troubleshooting DNS resolution problems.
- The traceroute utility, also known as tracert on Windows systems, uses ICMP to trace the path from one networked node to another, identifying all intermediate hops between the two nodes. This utility is useful for determining router or subnet connectivity problems.

- The World Wide Web (WWW, or Web) is a collection of internetworked servers that share resources and exchange information according to specific protocols and formats. On the client side, access to the Web requires the TCP/IP protocol, a unique IP address, a connection to the Internet, and a local interface to the Web called a browser.
- A Uniform Resource Locator (URL) that specifies the service used, its server's host name, and its HTML page or script name, identifies every Web page.
- Currently, e-mail is the most frequently used Internet service you will manage. Although e-mail packages vary in how they look and, to some degree, how they act, they all work on the same principles.
- The File Transfer Protocol (FTP) manages file transfers between TCP/IP hosts. FTP is a simple, yet important part of the TCP/IP suite. Before the WWW provided an easier means of transferring files, FTP commands were regularly used to exchange data between machines.
- Gopher is a text-based utility that allows you to navigate through a series of menus to find and read specific files. It is not sophisticated enough to interpret document formatting commands, such as HTML, but it does allow you to transfer files from one host to another by connecting with FTP.
- Newsgroups are similar to e-mail, in that they provide a means of conveying messages; they differ from e-mail, in that messages are distributed to a wide group of users at once rather than from one user to another.
- Rather than using SMTP (as e-mail does), newsgroup messages are transported by the Network News Transport Protocol (NNTP). NNTP supports the process of reading newsgroup messages, posting new messages, and transferring news files between news servers.
- The term "e-commerce" refers to a means of conducting business over the Web—be it in retailing, banking, stock trading, consulting, or training. Any buying and selling of products or services that occurs over the Internet belongs in the e-commerce category.
- An emerging Web-based service is Internet telephony, the provision of telephone service over the Internet. Given the Internet's breadth and economy, it seems logical that we would look to the Internet to carry the conversations that we currently transmit over PSTN. With a basic desktop computer equipped with a microphone, speaker, and the appropriate software, you could call anyone else with the same setup, essentially using your IP address as a telephone number.
- Today, significant technical obstacles prevent Internet telephony from becoming a widespread reality. First, voice conversations depend on the wire's quality of service. Also, voice transmissions are subject to distortion if the connection becomes too noisy. To compensate for delays, disorder, and distortion, a voice connection requires more dedicated bandwidth.

KEY TERMS

address resource record — A type of DNS data record that maps the IP address of an Internet-connected device to its domain name.

alias — A nickname for a node's host name. Aliases can be specified in a local host file.

anycast address — A type of address specified in IPv6 that represents a group of interfaces, any one of which (and usually the first available of which) can accept a transmission. At this time, anycast addresses are not designed to be assigned to hosts, such as servers or workstations, but rather to routers.

ARP table — The database that lists MAC addresses and their associated IP addresses used for ARP queries.

Bootstrap Protocol (BOOTP) — A service that simplifies IP address management. BOOTP maintains a central list of IP addresses and their associated devices' MAC addresses, and assigns IP addresses to clients when they request it.

browser — Software that provides clients with a simple, graphical interface to the Web.

convergence — The use of networks to carry data, plus video and voice signals.

core gateways — Gateways that make up the Internet backbone. The Internet Network Operations Center (INOC) operates core gateways.

default gateway — The gateway that first interprets a device's outbound requests, and then interprets its inbound requests to and from other subnets. In the postal service analogy, the default gateway is similar to a local post office.

diskless workstations — Workstations that do not contain hard disks, but instead rely on a small amount of read-only memory to connect to a network and to pick up their system files.

domain name — The symbolic name that identifies a domain. Usually, a domain name is associated with a company or other type of organization, such as a university or military unit.

Domain Name System (DNS) — A hierarchical way of tracking domain names and their addresses, devised in the mid-1980s. The DNS database does not rely on one file or even one server, but rather is distributed over several key computers across the Internet to prevent catastrophic failure if one or a few computers go down. DNS is a TCP/IP service that belongs to the Application layer of the OSI Model.

dotted decimal notation — The shorthand convention used to represent IP addresses and make them more easily readable by humans. In dotted decimal notation, a decimal number between 1 and 254 represents each binary octet. A period, or dot, separates each decimal.

dynamic address — An IP address that is assigned to a device through DHCP and may change when the DHCP lease expires or is terminated.

dynamic ARP table entry — A record (of an IP address and its associated MAC address) created in an ARP table when a client makes an ARP request that cannot be satisfied by data already in the ARP table.

Dynamic Host Configuration Protocol (DHCP) — An automated means of assigning a unique IP address to every device on a network.

echo reply — The response signal sent by a device after another device pings it.

echo request — The request for a response generated when one device pings another device on the network.

e-commerce — A means of conducting business over the Web — be it in retailing, banking, stock trading, consulting, or training. Any buying and selling of products or services that occurs over the Internet belongs in the e-commerce category.

extended network prefix — The combination of an address's network and subnet information. By interpreting an address's extended network prefix, a device can determine the subnet to which an address belongs.

File Transfer Protocol (FTP) — An Application layer TCP/IP protocol that manages file transfers between TCP/IP hosts.

Format Prefix — A variable-length field at the beginning of an IPv6 address that indicates what type of address it is (for example, unicast, anycast, or multicast).

gateway — A combination of hardware and software that enables one type of system to communicate with another type of system.

gopher — A text-based utility that allows you to navigate through a series of menus to find and read specific files.

host file — A text file that associates TCP/IP host names with IP addresses. On Windows 9x, NT, and 2000 platforms, the host file is called "lmhosts." On UNIX platforms the file is called "hosts" and is located in the /etc directory.

host name — A symbolic name that describes a TCP/IP device.

hosts — Name of the DNS host file found on a UNIX computer. The hosts file is usually found in the /etc directory.

Hypertext Markup Language (HTML) — The language that defines formatting standards for Web documents.

Hypertext Transport Protocol (HTTP) — The language that Web clients and servers use to communicate. HTTP forms the backbone of the Web.

ifconfig — A TCP/IP configuration and management utility used with UNIX systems (similar to the ipconfig utility used on Windows NT and 2000 systems).

Internet Mail Access Protocol (IMAP) — A mail storage and manipulation protocol that depends on SMTP's transport system and improves upon the shortcomings of POP. The most current version of IMAP is version 4 (IMAP4). IMAP4 can (and eventually will) replace POP without the user having to change e-mail programs. The single biggest advantage IMAP4 has relative to POP is that it allows users to store messages on the mail server, rather than always having to download them to the local machine.

Internet telephony — The provision of telephone service over the Internet.

ipconfig — The TCP/IP configuration and management utility for use with Windows NT or Windows 2000 systems.

IP next generation (IPng) — See *IP Version 6*.

IP version 6 (IPv6) — A new standard for IP addressing that will replace the current IP version 4 (IPv4). Most notably, IPv6 uses a newer, more efficient header in its packets and allows for 128-bit source and destination IP addresses. The use of longer addresses will allow for more total IP addresses to be in circulation.

lease — The agreement between a DHCP server and client on how long the client will borrow a DHCP-assigned IP address. As network administrator, you configure the duration of the lease (in the DHCP service) to be as short or long as necessary, from a matter of minutes to forever.

lmhosts — A host file on a Windows-based computer that maps IP addresses to host names and aliases.

multicast address — A type of address in the IPv6 that represents multiple interfaces, often on multiple nodes. An IPv6 multicast address begins with the following hexadecimal field: FF0*x*, where *x* is a character that identifies the address's group scope.

multicasting — A means of transmission in which one device sends data to a specific group of devices (not the entire network segment) in a point-to-multipoint fashion. Multicasting can be used for teleconferencing or videoconferencing over the Internet, for example.

name server — A server that contains a database of TCP/IP host names and their associated IP addresses. A name server supplies a resolver with the requested information. If it cannot resolve the IP address, the query passes to a higher-level name server.

name space — The database of Internet IP addresses and their associated names distributed over DNS name servers worldwide.

nbtstat — A TCP/IP troubleshooting utility that provides information about NetBIOS names and their addresses. If you know the NetBIOS name of a workstation, you can use nbtstat to determine its IP address.

netstat — A TCP/IP troubleshooting utility that displays statistics and the state of current TCP/IP connections. It also displays ports, which can signal whether services are using the correct ports.

network address translation (NAT) — A technique in which private (or hidden) IP addresses are assigned a public IP address by an IP gateway, thus masking their true origin.

Network News Transfer Protocol (NNTP) — The protocol that supports the process of reading newsgroup messages, posting new messages, and transferring news files between news servers.

Network Time Protocol (NTP) — A simple TCP/IP protocol that is used to synchronize the clocks of computers on a network. NTP belongs to the Application layer of the TCP/IP Model and depends on UDP.

newsgroups — An Internet service similar to e-mail that provides a means of conveying messages, but in which information is distributed to a wide group of users at once rather than from one user to another.

nslookup — A TCP/IP utility on Windows NT, Windows 2000, and UNIX systems that allows you to look up the DNS host name of a network node by specifying its IP address, or vice versa. This ability is useful for verifying that a host is configured correctly and for troubleshooting DNS resolution problems.

Packet Internet Groper (PING) — A TCP/IP troubleshooting utility that can verify that TCP/IP is installed, bound to the NIC, configured correctly, and communicating with the network. PING uses ICMP to send echo request and echo reply messages that determine the validity of an IP address.

pinging — The process of sending an echo request signal from one node on a TCP/IP network to another, using the PING utility.

port number — A unique number associated with a process running on a computer. For example, 23 is the standard port number associated with the Telnet utility.

Post Office Protocol (POP) — A TCP/IP subprotocol that provides centralized storage for e-mail messages. In the postal service analogy, POP is like the post office that holds mail until it can be delivered.

release — The act of terminating a DHCP lease.

resolver — Any host on the Internet that needs to look up domain name information.

resource record — The element of a DNS database stored on a name server that contains information about TCP/IP host names and their addresses.

Reverse Address Resolution Protocol (RARP) — The reverse of ARP. RARP allows the client to send a broadcast message with the MAC address of a device and receive the device's IP address in reply.

root server — A DNS server maintained by ICANN (in North America) that is an authority on how to contact the top-level domains, such as those ending with .com, .edu, .net, .us, and so on. ICANN maintains 13 root servers around the world.

Simple Mail Transfer Protocol (SMTP) — The TCP/IP subprotocol responsible for moving messages from one e-mail server to another.

socket — A logical address assigned to a specific process running on a host computer. It forms a virtual connection between the host and client.

static address — An IP address that is manually assigned to a device and remains constant until it is manually changed.

static ARP table entry — A record (of an IP address and its associated MAC address) that is manually entered in the ARP table using the ARP utility.

subnet mask — A special 32-bit number that, when combined with a device's IP address, informs the rest of the network as to what kind of subnet the device is on.

subnetting — The process of subdividing a single class of network into multiple, smaller networks.

switch — The letters or words added to a command that allow you to customize a utility's output. Switches are usually preceded by a hyphen or forward slash character.

top-level domain (TLD) — The highest-level category used to distinguish domain names—for example, .org, .com, .net. A TLD is also known as the domain suffix.

traceroute (or tracert) — A TCP/IP troubleshooting utility that uses ICMP to trace the path from one networked node to another, identifying all intermediate hops between the two nodes. Traceroute is useful for determining router or subnet connectivity problems.

Trivial File Transfer Protocol (TFTP) — A TCP/IP Application layer protocol that enables file transfers between computers. Unlike FTP, TFTP relies on UDP at the Transport layer and does not require a user to log onto the remote host.

unicast address — A type of IPv6 address that represents a single interface on a device. An IPv6 unicast address begins with either FFC0 or FF80.

Uniform Resource Locator (URL) — A standard means of identifying every Web page, which specifies the service used, its server's host name, and its HTML page or script name.

unqualified host name — A TCP/IP host name minus its prefix and suffix.

Voice over IP (VoIP) — The provision of telephone service over a TCP/IP network. (Pronounced "voyp".) One form of VoIP is Internet telephony.

well-known ports — TCP/IP port numbers 0 to 1023, so called because they were long ago assigned by Internet authorities to popular services (for example, FTP and Telnet), and are therefore well known and frequently used.

Windows Internet Naming Service (WINS) — A service that resolves NetBIOS names with IP addresses. WINS is used exclusively with systems that use NetBIOS—therefore, it is usually found on Windows-based systems.

winipcfg — The TCP/IP configuration and management utility for use with Windows 9x systems. Winipcfg differs from ipconfig in that it supplies a graphical user interface.

World Wide Web (WWW or Web) — A collection of internetworked servers that share resources and exchange information according to specific protocols and formats.

zone — The group of machines managed by a DNS server.

REVIEW QUESTIONS

1. How many octets are used for the network portion of a Class B IP address?
   a. 4
   b. 3
   c. 2
   d. 1
   e. 0
2. Which of the following dotted decimal addresses corresponds to the binary IP address 11111111 11111111 11111111 11111111?
   a. 10.10.10.10
   b. 100.100.100.100
   c. 127.0.0.1
   d. 255.255.255.255
   e. 1.1.1.1

3. What is another term for the address represented by 11111111 11111111 11111111 11111111?
   a. multicast address
   b. broadcast address
   c. loopback address
   d. qualified address
   e. RARP address
4. Why would a network manager choose to divide his or her TCP/IP networks into subnets?
   a. to conserve router interfaces
   b. to minimize traffic between segments on the network
   c. to use a limited number of IP addresses more efficiently
   d. to reduce the potential for IP addressing conflicts
   e. to enable a hierarchical addressing scheme for servers and workstations
5. What is the default subnet mask for a Class A network?
   a. 0.0.0.0
   b. 255.0.0.0
   c. 255.255.0.0
   d. 255.255.255.0
   e. 255.255.255.255
6. If a client workstation's IP address equals 119.55.60.122, and the network administrator is using subnetting, which of the following is probably the workstation's subnet mask?
   a. 0.0.0.0
   b. 255.0.0.0
   c. 255.255.0.0
   d. 255.255.255.0
   e. 255.255.255.255
7. Each node on a TCP/IP network has only one default gateway. True or False?
8. What is the primary advantage to using sockets?
   a. They enable clients and servers to communicate more expeditiously.
   b. They enable servers to keep a service available at all times.
   c. They eliminate the possibility that service requests might become corrupted.
   d. They ensure that error-correction is used during transmission.
   e. They ensure that a connection-oriented protocol is used when requesting services.

9. Which two ports are commonly used in FTP transmission?
   a. 23 and 24
   b. 21 and 22
   c. 20 and 21
   d. 22 and 23
   e. 24 and 25
10. Which top-level domain would the U.S. Congressional offices use?
   a. .loc
   b. .info
   c. .com
   d. .gov
   e. .mil
11. You are the network manager for a realtor with 50 workstations in its TCP/IP-based LAN. One day your supervisor tells you that the company is doubling in size in the next month and that you will have to add 50 new workstations to the network in the next two weeks, making sure they can use both internal and external resources. However, you realize that you only have enough Internet-authorized IP addresses for the existing 50 workstations and cannot get new IP addresses in time for setting up the new workstations. Which of the following is the best solution to this problem?
   a. Create new lmhosts files for each Windows-based system that map the IP address of each other system to its MAC address.
   b. Assign each of the 50 IP addresses to two workstations and configure the border router to allow only one of those workstations to access a public network at any time.
   c. Call a local ISP and ask to lease another 50 IP addresses until you can reserve your own through ICANN.
   d. Establish a peer-to-peer network that enables the 100 workstations to share the 50 legitimate IP addresses.
   e. Set up an IP gateway that can perform network address translation and assign the new workstations addresses that can only be used within the organization.
12. What three columns of information would you find in a file called "lmhosts" on a Windows 2000 workstation?
   a. IP address, domain name, and NetBIOS address
   b. IP address, NetBIOS address, and alias
   c. IP address, domain name, and alias
   d. IP address, host name, and alias
   e. IP address, MAC address, and host name

13. How can a resolver quickly find addresses for previously visited sites?
14. Which of the following symbols indicates a comment in an lmhosts file?
    a. $
    b. &
    c. ^
    d. /
    e. #
15. An organization may use only one name server. True or False?
16. What is the significant disadvantage to using BOOTP?
    a. It requires two servers to hold the information for a single subnet.
    b. Its tables must be manually updated.
    c. It does not conform to the hierarchical DNS model.
    d. It is unreliable.
    e. It is not supported by newer operating systems.
17. Although DHCP enables users to move their machines from one location on the network to another without reconfiguring TCP/IP settings, what might a user have to do once she has moved to a new part of the network so as to use TCP/IP applications?
    a. change her primary DNS setting
    b. change her default protocol setting
    c. release and renew her IP address
    d. adjust the time on her DHCP lease
    e. change her SNMP server name
18. When a client needs to obtain an IP address from a DHCP server, what kind of transmission does it send?
    a. broadcast UDP
    b. multicast TCP
    c. broadcast RARP
    d. multicast RARP
    e. unicast UDP
19. In total, how many packets are exchanged in the process of a client requesting and obtaining an IP address from a DHCP server?
    a. 2
    b. 4
    c. 6
    d. 8
    e. 10

20. Which of the following is a benefit of using WINS?
    a. It ensures that every device on the network has a unique IP address.
    b. It ensures that every device on the network has a unique host name.
    c. It ensures that every device on the network has a unique NetBIOS name.
    d. It ensures that every device on the network has a unique socket address.
    e. It ensures that every device on the network has a unique port address.
21. What is the primary difference between UDP and TCP?
    a. UDP is a Data Link layer protocol and TCP is a Network layer protocol.
    b. UDP is connectionless and TCP is connection-oriented.
    c. UDP relies on IPX/SPX and TCP relies on TCP/IP.
    d. UDP is more secure than TCP.
    e. UDP does not require IP while TCP does.
22. How could RARP benefit diskless workstations?
23. What is SMTP's primary function?
    a. to provide management information about network devices
    b. to monitor security breaches at the router interface level
    c. to transport mail from one host to another
    d. to supply port usage information
    e. to automatically issue network alerts when the nameserver goes down
24. Where does mail go after it is retrieved by an e-mail program that uses POP?
    a. to the user's mail directory on the mail server
    b. to the user's mail directory on the client workstation
    c. to the root directory on the client workstation
    d. to the recycle bin on the mail server
    e. to an archive disk on the mail server
25. Which two of the following are benefits of using IMAP4 relative to POP?
    a. It provides mail delivery guarantees.
    b. It allows users to review and delete mail without downloading it.
    c. It allows users to create mail messages on the server.
    d. It provides better encryption for message attachments.
    e. It enables multiple users to easily share a central mailbox.
26. IMAP4 can work without SMTP, but POP3 cannot. True or False?

27. Why is it critical that the clocks of all computers on a network are synchronized?
    a. so that multiple servers on a network can manage shared, external storage devices without, for example, overwriting the most recent version of a file with an older version of the file
    b. so that DHCP servers do not inadvertently assign the same IP address to two different devices
    c. so that packets are not prematurely discarded due to a difference in time between the source node and the devices between the source and target nodes
    d. so that data backups will not begin while users are still connected to the network, potentially opening files and making them vulnerable to corruption
    e. so that the DHCP server can accurately track which devices have leased IP addresses that will soon expire
28. What can you learn by pinging the loopback address?
29. If you know that your boss's TCP/IP host name is JSMITH, and you need to find out what her IP address is, what command (with correct syntax) should you type at your DOS prompt?
    a. `nslookup jsmith`
    b. `netstat jsmith`
    c. `tracert jsmith`
    d. `whois jsmith`
    e. `nbtstat jsmith`
30. What command might you use to find out whether your ISP's router is especially slow on a particular afternoon?
    a. `traceroute`
    b. `nbtstat`
    c. `netstat`
    d. `nslookup`
    e. `ipconfig`
31. How can you view a list of FTP commands once you have connected to an FTP server?
    a. type: `commands`
    b. type: `show`
    c. type: `quit`
    d. type: `help`
    e. type: `Q`

32. What command would you use to list the TCP/IP configuration for all three of the NICs in your Linux server?
    a. `nslookup –all`
    b. `ipconfig –all`
    c. `ipconfig /all`
    d. `ifconfig –a`
    e. `nslookup -a`
33. Which protocol is used to transmit and receive messages to and from newsgroups?
    a. NNTP
    b. SNMP
    c. NCP
    d. SMTP
    e. NSFT
34. Why is security a concern for Web sites that offer e-commerce?
35. What type of device digitizes, compresses, and assembles into packets a voice phone call, which it receives from a telephone switch?
    a. TCP/IP switch
    b. NIC
    c. IP telephone
    d. name server
    e. gateway

HANDS-ON PROJECTS

Project 11-1

In this exercise, you will set up a Windows 2000 workstation with everything it needs to access the Internet. For this project, you will need a Windows 2000 workstation that currently has TCP/IP installed and bound to the NIC, but doesn't have any settings specified. You will need to obtain the correct settings for your network from your instructor. In this project, it's important to type the numbers exactly as they are given to you; otherwise, the TCP/IP connection will not work. Each student should use a unique IP address and host name.

1. Obtain the following numbers from your instructor: IP address, subnet mask, DNS primary name server, DNS secondary name server, default gateway, and domain name.
2. Click **Start**, point to **Settings**, and then click **Control Panel**.

3. Double-click **Network and Dial-up Connections**. The Network and Dial-up Connections window opens.
4. Right-click the **Local Area Connection** icon, then click **Properties** in the shortcut menu. The Local Area Connection Properties dialog box appears.
5. Select **Internet Protocol (TCP/IP)** in the list of installed network components, then click **Properties**. The Internet Protocol (TCP/IP) Properties dialog box opens.
6. Make sure the **Use the following IP address** option is selected.
7. Type your network's IP address in the space provided.
8. Type your network's subnet mask in the space provided.
9. Type your network's default gateway in the space provided.
10. Make sure the **Use the following DNS server addresses** option is selected.
11. Enter the IP address of your primary DNS server in the **Preferred DNS server** text box.
12. Enter the IP address of your secondary DNS server, if you have one, in the **Alternate DNS server** text box.
13. Click **OK** to save your changes to the TCP/IP properties. The dialog box closes.
14. Click **OK** to save the changes to the Network properties. The dialog box closes and you are prompted to restart your computer to save your changes.
15. Click **Yes** to confirm that you want to restart your computer. When your workstation restarts, it will be properly configured for TCP/IP to browse the Web and to use TCP/IP applications such as FTP and Telnet.
16. To test whether your change worked, click **Start**, point to **Programs**, point to **Accessories** point to, and then click **Command Prompt**. The Command Prompt window opens.
17. At the command prompt, type `telnet locis.loc.gov` and press **Enter**.
18. If you see a text screen entitled "LOCIS: Library of Congress Information System," you have successfully modified your TCP/IP properties. If you see a window titled Connect Failed!, you either typed the host name incorrectly or need to retrace your steps from the beginning of this exercise to ensure that your TCP/IP properties are correct.
19. Following the instructions on the screen, logout from the Library of Congress system, then close the Command Prompt window.

Project 11-2

Computer scientists around the world collaborate to devise Internet protocols and standards. These standards, along with comments and Internet-related meeting notes, are then transformed into Requests for Comments (RFCs). When you want to find the source of an Internet standard, you can look in its RFCs. Some RFCs were written at the genesis of the Internet and have since been revised several times. New RFCs are continually being written. In this exercise, you will use an FTP client to find RFCs at different Internet host sites and explore their content.

For this and subsequent exercises, you will need a PC workstation with access to the World Wide Web (a standard TCP/IP installation and an Internet connection such as the one you configured in Project 11-1). Your workstation should also have a Web browser, such as Internet Explorer or Netscape Communicator, installed.

1. Verify that your workstation is connected to the Internet.
2. At the command prompt, type **`ftp`** and press **Enter** to begin an FTP session.
3. At the ftp prompt, type **`help`** and press **Enter**. How many FTP commands does your FTP client provide? Do any of the commands look familiar?
4. Type **`open ftp.isi.edu`** and press **Enter** to connect to the University of Southern California Information Sciences Institute FTP site, where an official record of RFC documents is kept.
5. Now you need to enter your user name. Because this site allows guests to log in with the user name "anonymous," type **`anonymous`** and press **Enter**. (Because the user name is case-sensitive, make sure you don't type any capital letters.)
6. Now you need to enter a password. Type your e-mail address as your password, and then press **Enter**. If you do not have a valid e-mail address, ask your instructor to provide an address you can use for this purpose.
7. To confirm that you have logged in, the ISI FTP server greets you with a long message that begins: "Guest login ok, …."
8. To change directories to the folder that contains the RFC documents, type **`cd in-notes`** at the ftp prompt, and then press **Enter**. This command is case-sensitive, so be sure not to use any capital letters.
9. To show a listing of all RFCs in this directory, type **`ls`** and press **Enter**. Because there are so many RFC documents, this listing will take a while to complete.
10. To copy RFC number 1816 to your hard disk, type **`get rfc1816.txt "c:\rfc.txt"`** and press **Enter**. Note that "get" is the FTP command for retrieving a file. The name of the file on the FTP server is "rfc1816.txt" and "c:\rfc.txt" is the filename you will save it under on your computer. Also note that the default file transfer mode is ASCII, which is appropriate because the RFC is a text file.
11. Open the file c:\rfc.txt using a text editor program (if you are working on a Windows workstation, you can use WordPad).
12. Read the header and at least a few paragraphs from this RFC. What is the topic of this RFC? What previously written RFC does it replace? On what date was it published?
13. Repeat Step 10, but rather than retrieving RFC 1816, retrieve RFC 2146, saving it to a file named c:\rfc2.txt. Open the file in a text editor program and note how it pertains to RFC 1816.

14. Now repeat Step 10 to retrieve another RFC, this time RFC 2151, saving it to a file named c:\rfc3.txt. Review this file in a text editor program. How much of it looks familiar? What new information can you learn from this document?
15. To close the FTP session, type `quit` and press **Enter**. Close the Command Prompt window.

Project 11-3

In this exercise, you will use a Web browser to locate an RFC. In addition, you will perform Web searches and use services other than HTTP. To complete this project, you need a workstation with Internet access.

1. Verify that your workstation is connected to the Internet.
2. Go to the following Web site: *www.rfc-editor.org/rfcsearch.html*.
3. Under the Search text box, type **1816**, and then click the **SEARCH** button.
4. A list of matching RFC documents appears in the bottom half of the screen. Click the number **RFC1816** to open this document in your browser window.
5. According to this RFC, which domain suffix should local and state agencies use? (*Hint*: If you are reading the document in a browser window, you can use your browser's search function to look for any instance of the phrase "local and state.")
6. Go to the Internet Engineering Task Force's home page: *www.ietf.org*.
7. To view a list of the IETF's working groups, click the **IETF Working Groups** link.
8. Scroll through the page to get an idea of how many emerging Internet technologies need to be standardized or refined. Under the Operations and Management Area, choose to view more information about the Remote Network Monitoring group. What is the purpose of this group? What RFCs have resulted from its work?
9. Go to the following Web site: *ftp://ftp.isi.edu/in-notes/*. Look at the list of text files and directories that appear. It is the same list that you viewed through the FTP command `ls` in Project 11-2.
10. Type **`altavista`** in your Web browser's address box, press **Enter**, and see whether your browser adds the prefix and suffix to the unqualified host name and finds the appropriate site. If it doesn't, point your browser to the fully qualified host name, *www.altavista.com*.
11. At the AltaVista search engine site, type the following question in the search field: **`What is SMTP`**? Click **Search**.
12. What kind of response do you get from the search engine?

Project 11-4

In this project, you will gain more experience with simple TCP/IP troubleshooting commands. To complete this project you will need a Windows 2000 Professional workstation that can connect to the Internet.

1. Connect to the Internet, click **Start**, point to **Programs**, point to **Accessories**, and then click **Command Prompt**. The Command Prompt window appears.
2. At the command prompt, type **`netstat –a`** and press **Enter**. Recall that **`netstat`** is the command that reveals all TCP/IP port connections, even if they are not actively exchanging data. How many connections are listed on your computer? Of those connections, how many rely on the TCP protocol and how many rely on UDP?596
3. Now look at the "State" column of your connection listing. How does the value in this column differ for TCP and UDP connections? Why do you suppose this is the case?
4. Now type **`netstat –s`** and press **Enter**. How many different TCP/IP core protocols are currently in use on your machine? Of those, which one has sent and received the most packets?
5. Now you will experiment with another TCP/IP utility, the traceroute function. At the command prompt, type **`tracert www.course.com`** and press **Enter**. How many hops does it take to go from your computer to the Course Technology home page's computer? How many hops are listed as the maximum for the tracert command?
6. Now use the traceroute utility but allow it to omit the host names of every hop between your workstation and the destination by typing **`tracert –d www.course.com`** and pressing **Enter**. Notice how the output differs from the output you received in Step 5. Close the Command Prompt window.

Project 11-5

In this exercise, you will configure an e-mail program with the correct mail server settings. In a typical LAN environment, users will not know how to specify their mail servers or whether their software uses POP or IMAP4. In addition to knowing this basic information, you should be familiar with any special mail program settings required by your organization, such as whether users are allowed to leave messages on the server and, if so, how long they are allowed to maintain them there.

For this exercise, you will need a Windows 9x or Windows 2000 Professional workstation with one of the popular e-mail programs, such as Eudora Light, installed. If you do not have the disks to install Eudora Light, you can download the software from Qualcomm's Web site at *www.eudora.com/products/eudora/download/* at no cost. Your instructor can provide the configuration information for your network, such as the mail server name, if you do not already know it.

Although this exercise focuses on the configuration process for the Eudora e-mail package, all Internet e-mail packages will require you to enter the same information, if not more. Many e-mail programs share similar menu structures, too. Once you have experience configuring a few popular programs, therefore, you can easily figure out how to configure other e-mail programs.

1. To run the Eudora Light mail program, click **Start**, point to **Programs**, point to **Eudora**, and then click **Eudora**. The Eudora window opens.
2. Click **Tools** on the menu bar, and then click **Options**. The Options dialog box opens, as shown in Figure 11-31.

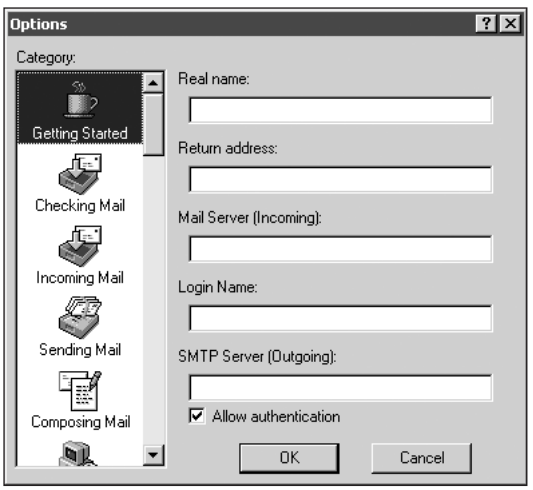

Figure 11-31 Eudora Options dialog box

3. Scroll to the top of the Category list box, and then click the **Getting Started** icon.
4. In the Real name text box, type your first and last name.
5. In the Return address text box, type your full e-mail address.
6. In the Mail Server (Incoming) text box, type your mail server name. In the Login Name text box, type your e-mail account name. In the SMTP Server (Outgoing) text box, type your SMTP server name (which may be the same as your mail server name).
7. In the Category list box, click the **Sending Mail** icon.
8. In the Domain to add to unqualified addresses text box enter your mail server's domain name. For example, if your e-mail address is jdoe@usa.com, enter usa.com. This choice enables you to send mail to your friend usam@usa.com by typing simply usam in the To: field of your e-mail message.
9. In the SMTP server text box, enter your SMTP server name.
10. Save your settings by clicking **OK**. You return to the main Eudora Light window.

11. To test your mail client, click **File** on the menu bar, and then click **Check Mail**. Why didn't you have to restart your workstation for these changes to take effect?
12. In the next steps, you will see what happens when a mail server name is spelled incorrectly. To start, click **Tools** on the menu bar, and then click **Options**.
13. Click the **Sending Mail** icon. The sending mail options appear.
14. Change one letter in the name of your SMTP server, and then click **OK** to save your changes.
15. To create a test message to send to yourself, click **Message** on the menu bar, and then click **New Message**. A new message window appears, prompting you to enter the recipient's e-mail address.
16. Type your own e-mail address (correctly) in the To text box.
17. In the Subject: text box, type **`test`**.
18. Click the **Send** button. What happens? What kind of error message do you receive when you attempt to retrieve your mail?
19. Follow Steps 13 through 17 again, this time typing in your correct SMTP server name to restore your ability to send e-mail from the Eudora Light client.

CASE PROJECTS

1. Katie Stark, who owns a local greenhouse called Katydid Nursery, knows you from college. She has heard you're a networking expert and calls to ask how she can sell bulbs, seeds, garden tools, and houseplants on the Web. To date, Katie has used computers only to keep her inventories, but she's heard that e-commerce is an easy way to make more money. Her greenhouse employees use five computers, all Pentium IIIs. She doesn't think that they are even connected to the Internet. Katie has a limited budget and frankly isn't sure whether having a Web site is something she can afford. If she had a Web site, she tells you, she would call it www.katydids.com. Based on what you know about Internet connections, Web sites, and e-commerce, what kind of connection do you recommend? What advice can you provide about establishing a Web site and preparing her computers to use the Web?

2. Katie told her friend Andy about your skills with computers. Andy knows a bit about computers himself, as he is the sole network technician at a chain of six hardware stores in the Pacific Northwest. His network is connected to the Internet through a dedicated T1 link, and he has registered the address range from 205.38.123.1 through 205.38.123.100 with ICANN. For the most part, the clients Andy supports use the Internet for e-mail, Web surfing, and exchanging files. The problem with his network is that he's the only employee who knows anything about TCP/IP, and the network is growing faster than he can handle. Andy is constantly visiting one or another of his users' machines to check their IP address in their TCP/IP properties window, or finding that they've accidentally deleted their gateway setting. How can you help him?
3. Andy's surprised that you know so much about TCP/IP. On a day when his connection to the Internet is extremely slow, he challenges you to find out why the performance is so bad. Andy jokes that it might have something to do with the earthquake that just took place in San Jose. What do you do?